UNIVERSITY OF ALBERTA

INTEGRATING A DESKTOP GEOGRAPHIC INFORMATION SYSTEM (GIS) WITH A LOCATION-ALLOCATION MODEL

BY

ROD SCHATZ ©

A thesis submitted to the Faculty of Graduate Studies and Research in partial fulfillment

of the requirements for the degree of **Master of Science.**

Department of Earth and Atmospheric Sciences

Edmonton, Alberta

Spring 2000

National Library
of Canada

Acquisitions and
Bibliographic Services

395 Wellington Street
Ottawa ON K1A 0N4
Canada

Bibliothèque nationale
du Canada

Acquisitions et
services bibliographiques

395, rue Wellington
Ottawa ON K1A 0N4
Canada

Your file Votre référence

Our file Notre référence

The author has granted a non-exclusive licence allowing the National Library of Canada to reproduce, loan, distribute or sell copies of this thesis in microform, paper or electronic formats.

The author retains ownership of the copyright in this thesis. Neither the thesis nor substantial extracts from it may be printed or otherwise reproduced without the author's permission.

L'auteur a accordé une licence non exclusive permettant à la Bibliothèque nationale du Canada de reproduire, prêter, distribuer ou vendre des copies de cette thèse sous la forme de microfiche/film, de reproduction sur papier ou sur format électronique.

L'auteur conserve la propriété du droit d'auteur qui protège cette thèse. Ni la thèse ni des extraits substantiels de celle-ci ne doivent être imprimés ou autrement reproduits sans son autorisation.

0-612-60185-4

Canada

UNIVERSITY OF ALBERTA

LIBRARY RELEASE FORM

Name of Author:	**Rod Schatz**
Title of Thesis:	**Integrating a Desktop Geographic Information System (GIS) with a Location- Allocation Model**
Degree:	**Master of Science**
Year this Degree Granted:	**2000**

Permission is hereby granted to the University of Alberta Library to reproduce single copies of this thesis and to lend or sell such copies for private, scholarly, or scientific research purposes only.

The author reserves all other publication and other rights in association with the copyright in the thesis, and except hereinbefore provided, neither the thesis nor any substantial portion thereof may be printed or otherwise reproduced in any material form whatever without the author's prior written permission.

Rod Schatz

12 White Oak Place
St. Albert, Alberta T8N 3H4

Date: April 4th · 2000

UNIVERSITY OF ALBERTA

FACULTY OF GRADUATE STUDIES AND RESEARCH

The undersigned certify that they have read, and recommend to the Faculty of Graduate Studies and Research for acceptance, a thesis entitled **INTEGRATING A DESKTOP GEOGRAPHIC INFORMATION SYSTEM (GIS) WITH A LOCATION-ALLOCATION MODEL** submitted by **Rod Schatz** in partial fulfillment of the requirements for the degree of **Master of Science**.

Dr. M. John Hodgson

Erhan Erkut

Dr. Erhan Erkut

Gilles Reinhardt

Dr. Gilles Reinhardt

Date: November 15, 1999

Abstract

This research integrates a Geographic Information System (GIS), Relational Database Management System (RBDMS) and Location-Allocation (LA) into a system that alleviates data handling problems in locational analysis. Custom developed functions are incorporated into an ArcView® GIS add-on module known as ArcView's® Location-Allocation (ARCLA) modelling extension. ARCLA simplifies many of the tedious data processing tasks such as data query, display and manipulation. ARCLA works in conjunction with a RDBMS to provide real world data sets to use with LA models and socio-economic data analysis. The RDBMS stores the 1996 Census of Population data for all Census Metropolitan Areas (CMA) in Canada. The use of this system demonstrates that GIS is very beneficial to LA modelling. Using this system, the effects of data surrogation error are investigated for fifteen metropolitan Canadian cities. The surrogation error results reveal that surrogation error is always present, albeit in some instances it may be minimal.

Dedication

This thesis is dedicated to my parents for their wonderful support during my lengthy academic pursuits and to my fiancée, Stacy, who has always been a wonderful source of encouragement and counsel.

Acknowledgments

I would like to thank a number of people who provided assistance during the development of this research study. Heidi Grainger and Stacy Grainger for their fantastic editorial skills. I would also like to congratulate Stacy for the success of her own dissertation in the Fall of 1998. Al Muirhead and Darren Wolchyn of Harrison Muirhead Systems Inc. for their guidance with many of technical database issues encountered during the implementation of this research. Arndt Buhlmann for his wonderful listening skills as I passed ideas by him during the many stages of this research. Alex Miller of ESRI Canada for donating the Programming with Avenue (advanced course) notes, which proved to be a tremendous resource during the development of the ArcView Extension. GDS & Associates Systems Inc., my present employer, for their tremendous understanding and support to ensure that I was able to finish this research in a timely manner. Without the assistance of everyone mentioned above this research would not have come to fruition.

TABLE OF CONTENTS

Chapter **Page**

I. Introduction 1

Thesis Outline 3

II. Background to the Study 5

Classic Location-allocation models 5

Real world data structure 9

Geographic Information Systems and model integration 10

Relation of this Study to the Literature 13

III. Research Problem 15

Research Problem 15

GIS Fundamentals 15

Location-Allocation Model 18

Location-Allocation Study 19

Limitations of this research 19

VI. 1996 Census of Population Data 21

Database Fundamentals 21

1996 Census of Population Data 26

Census of Population - Attribute Data 27

Development of a Census of Population Attribute Database 28

1996 Spatial Centroid Data 35

Census of Population - Spatial Databases 36

Census of Population - Attribute Data Quality, Sampling, Weighting And Random Rounding 40

Census of Population - Spatial Data Quality 41

CMA Spatial Databases - Metadata 42

Summary 42

V. Integrating GIS and Location-Allocation Models 43
ArcView® GIS Overview 44
ArcView® Views 45
ArcView® Tables 46
ArcView® Charts 47
ArcView® Layouts 47
ArcView® Scripts 47
Conceptual Integration Design 48
Location – Allocation Files 50
ARCLA Basics 53
Census Attribute Database Integration 54
Creating the Demand Nodes File 56
Creating the Potential Facilities Nodes File 56
Creating the Location-Allocation Parameter Files 57
Executing the Location - Allocation Program 57
Mapping Location-Allocation Solutions in ArcView® 58
ARCLA Utilities 60
Enhancing ARCLA 63
Summary 65
VI. Location-Allocation Analysis 66
The Location-Allocation Model 68
Location-Allocation Problem Setup 69
GIS Analysis 71
Surrogation Error 77
Summary 81
VII. Summary and Discussion 83

First Objective 83

Second Objective 84

Data Analysis Section 86

Applications of the GIS/LA System 87

Conclusion 87

Further Discussion 88

VIII. Bibliography 89

Appendix 1 92

Appendix 2 127

Appendix 3 134

Appendix 4 137

Appendix 5 144

Appendix 6 165

Appendix 7 170

Appendix 8 179

LIST OF TABLES

Tables **Page**

Table 4-1: Demographic variables covered by each Nation Table 32

Table 4-2: Primary CMA regions with their respective population and EA zone values 38

Table 4-3: Secondary CMA regions with their respective population and EA zone values 39

Table 5-1: Location-allocation program input files specifics 51

Table 5-2: Location-allocation program solution files specifics 53

Table 6-1: Population breakdowns for the fifteen selected Canadian cities 69

Table 6-2: Location-allocation program set up statistics 70

LIST OF FIGURES

Figures **Page**

Figure 4-1: Organization of data as records in a table 22

Figure 4-2: Relational join and query results 24

Figure 4-3: Sample entity relationship diagram for a Customer Order database 25

Figure 4-4: EAuid lookup table - screen capture of a small section of the table 31

Figure 4-5: Entity relationship diagram for the final 1996 Census Attribute Database design 34

Figure 4-6: City of Edmonton - EA zones with representative points (centroids) 36

Figure 4-7: Digital Cartographic File for Victoria, British Columbia at the EA level 37

Figure 4-8: CMA region for Vancouver, British Columbia at the EA level 40

Figure 5-1 ArcView's Graphical User Interface 45

Figure 5-2: Conceptual Integration design between the Census Attribute Database, ArcView and the location-allocation program 49

Figure 5-3: ArcView's location-allocation extension ARCLA interface 54

Figure 5-4: Demonstration of the final output from the ARCLA extension 63

Figure 6-1: Senior's population per capita in Edmonton 72

Figure 6-2: Children's population per capita in Edmonton 72

Figure 6-3: Total population distribution in Edmonton 73

Figure 6-4: Final facility allocations for seven preschool children facilities (less than six years of age) in Edmonton 74

Figure 6-5: Final facility allocations for seven senior facilities (65 years of age and older) in Edmonton 75

Figure 6-6: Final facility allocations for seven total population facilities in Edmonton 76

Figure 6-7: Cost-Effectiveness Graph for Edmonton 77

Figure 6-8: Location Error - children and seniors (proper population) demographic groups compared to the total population (surrogate data) for Edmonton 79

Figure 6-9: Surrogation Error - seniors (proper population) demographic group compared to the total population (surrogate data) for all fifteen cities 80

Figure 6-9: Surrogation Error - children (proper population) demographic group compared to the total population (surrogate data) for all fifteen cities 81

Chapter I. Introduction

Over the last 30 years, location analysts have developed methods and models to solve complex problems concerned with determining the optimal location of facilities. These models, often termed location-allocation (LA) models, simultaneously locate several facilities and allocate demands, expressed as weights at nodes, to them. The overall intention of LA models is to optimally serve the demands at each node, which in turn are allocated to a facility. The complexity of these models has led the location analysis community to model problems about real world situations using simplified means. One such means, employed by many of these practitioners, is to simplify their real world surroundings by using randomly generated data sets to represent random locations and random demand weights. Even today, many in the LA community continue to use randomly generated data sets in place of real world conditions to create, formulate and test new location models.

To truly understand a location model, it is argued that it must be tested with objective, non-structured data, since these models are designed to predict the real world scenarios. Many of these researchers do not utilize real world data sets as the data possess an inherent structure, which can bias the results of their research. An example of such structure is the dense concentration of people closer to the central business district (CBD), and the thinning out of people towards the municipal boundary of many North American cities. Studies that use real world data typically only perform analysis on one city and this can bias their results because of the city's inherent structure. Moreover, real world data involves the time consuming tasks of data collection and manipulation. In addition, many researchers believe that the time they would spend collecting and developing data sets would be better served by continuing their research. Another associated issue is the costliness of data. Data are very expensive to purchase and consequently, it is simply cheaper to write algorithms that generate random numbers. Hodgson, Rosing, and Storrier (1997) concluded that real-world applications require a large amount of data manipulation and that this was the most time consuming element of their research which tested a bicriterion LA model on a real world road network.

Geographic Information Systems (GIS) introduced in the 1960s have become an important technology for the integration of data from different sources and expertise from different disciplines (Burrough & McDonnell, 1986, p. 15). The potential for GIS benefiting LA modelling is substantial. At the present moment, the GIS field could make excellent use of the analytical abilities of the LA field. Conversely, the location-allocation field could employ many of the advancements that the GIS field has made in the last few years to simplify data management and display. Presently a dichotomy exists as both research disciplines could utilize the abilities of the other, yet the integration of the two fields is early in the evolutionary process.

Therefore, there is a need to investigate methods by which the location-allocation and the GIS fields could be integrated to take advantage of the strengths of both fields. This research is a direct consequence of the conditions stated above and seeks to integrate the two fields and to elevate data availability and manipulation issues for LA researchers. Consequently, the major goal of this research is to alleviate the data handling problems by providing the location researcher with tools to facilitate quick access to real world data sets, which are useful for LA analysis. These tools are also very useful to researchers in other disciplines who deal with socio-economic related problems. Thus, the objectives of this study are:

(1) to demonstrate the ease with which a desktop GIS and a Relational Database Management System (RDBMS) can be integrated to provide quick and efficient access to real world data. The purpose of the database is to serve as a data bank or warehouse of socio-economic variables that can be linked to GIS data sets to be used in LA analysis and problems involving the analysis of socio-economic data.

(2) to develop a system that integrates a desktop Geographic Information System (GIS) and a LA model in such a manner that the integration takes advantage of the strengths of both applications.

After the two objectives for this study have been achieved, a location-allocation study is undertaken using real world data for fifteen Canadian cities to look at the effects of surrogation error. LA analysis using fifteen cities would have been very difficult to do before without the aid of the tools developed for this study. This section of the study demonstrates the reconciliation of the dichotomy by not using random numbers and the results are not biased by one city's structure because the results of all fifteen cities are combined. The results determine the viability of using a real world GIS/LA system for solving LA problems. This component of the research is done using the tools set forth above and demonstrates that real world data are easy to work with.

Thesis Outline

Following this brief introductory chapter, the thesis is broken down into several major components. In Chapter 2, the literature review sets the stage for the study, by reviewing the work that has been done to date and the relationship of this work to the present study. First, the chapter discusses classic LA models from a rudimentary standpoint on how the models function and how the solution results are displayed. Next, there is a brief discussion on LA researchers who are presently exploring the use of a GIS with LA modelling. The final section of Chapter 2 discusses the integration of LA models and GISs.

The next chapter describes the research problem and methodology that is used in the present study. This chapter introduces the two types of data, spatial and attribute data that are available from Statistics Canada for the 1996 Census of Population. The objective of this chapter is to explain how the available Census of Population data can be modified and developed in a database that provides quick access to the socio-economic data.

Chapter 4 presents the background information on database management systems and demonstrates simple database techniques that allow for the development of a Census Attribute Database. The Census Attribute Database is created using data from the 1996 Canadian Census of Population. This chapter addresses part of the first objective of the study.

The second objective is addressed in the next chapter, Chapter 5, which deals with developing a system that integrates a GIS software package, the Census Attribute Database and a LA model. This chapter explains how a LA model and a GIS can be integrated and then describes the GIS tools that can be developed to simplify data manipulation and display problems for LA modelling. The chapter also explains how the Census Attribute Database can be integrated with a GIS to provide quick and efficient access to socio-economic variables for LA analysis and the database provides data sets for analysis of other socio-economic problems, thus fulfilling the first and second objectives of this study.

Chapter 6 concentrates on the dichotomy discussed before by performing LA analysis using real world data for fifteen Canadian cities. This chapter is the most important chapter of the present study because it demonstrates the ease with which a real world system can be used for LA analysis. This chapter presents the results of a LA study using the GIS/LA tools that were developed in the preceding chapters. The socio-economic data stored in the Census Attribute Database are used for the LA analysis section. The analysis utilizes the GIS tools developed and described in Chapter 5. The final chapter presents the main conclusions that can be drawn from the present study.

The GIS/LA system that is developed for this research is very useful for answering questions about location and the socio-economic conditions present in each of Canada's Census Metropolitan Areas. The GIS/LA system is useful for enhancing analysis performed by location-allocation practitioners, business analysts, demographers, urban planners and social scientists.

Chapter II. Background to the Study

Classic Location-allocation models

Location theory has taken many different forms since its beginnings. Location studies were founded by people such as the German economist Von Thünen (1826) who based his analysis on the location of economic rent and its characteristics to the German countryside. Another German economist, Weber (1909) looked at identifying the least transportation cost point on a surface for a single raw material. In essence, all of these models are normative and draw heavily upon neo-classical micro-economic theory (Pocock & Hudson 1978). Since these early developments in the location field, research has concentrated on models which seek to determine the optimal location of discrete facilities. The LA field expanded immensely in the mid-twentieth century as new models and solution techniques were being developed.

Lea (1978) summarized the essential nature of location-allocation problems by stating that all of these problems have the following features: "...a number of discrete facilities providing one or more goods or services that are to be located in geographic space to optimize an objective function subject to constraints. In addition to finding a system of locations, it is required to optimally allocate a set of spatially distributed users, consumers or suppliers to the facilities..." (Lea, 1978, p. 166).

One LA model in particular, the *p*-median model, is a problem with a number of demand and candidate locations such that the distance within the system is minimized. The objective of the *p*-median problem is to locate a given number of facilities, so that the facilities serve the required number of demand points by minimizing the aggregate weighted distance between each demand point and the facilities to which they are allocated. The *p*-median model assumes that each demand node is completely served by the facility located closest to it. Since the model strives for minimization, it forces all demand allocations to the closest facility. The specific target groups can be identified by specific socio-economic variables; therefore, socio-economic variables can be used to define the targets groups. For example, the *p*-median model could be used to determine the location for one or more daycare facilities in a certain city. This would be accomplished by using socio-economic variables that describe where preschool children reside and how many reside in each area.

A distance matrix is used to store the distance measurement between each origin and destination node. There are several ways of measuring distance. Two distance measurements are Cartesian distance and Great Circle distance. Cartesian distance is used to calculate distance on a two-dimensional plane and on a sphere, such as the Earth, Great Circle distance is used.

On a Cartesian plane, which is a two-dimensional surface where features (points, lines and polygons) are defined by a set of x and y coordinates, the distance between two points is measured as Euclidean or "straight line" distance. This distance measure is good for two-dimensional surfaces but is not accurate for three-dimensional surfaces. Euclidean distance is evaluated with the following formula:

$$D_{i,j} = \left((X_i - X_j)^2 + (Y_i - Y_j)^2 \right)^{1/2}$$

On a sphere, such as the Earth, straight-line distances are not applicable, instead; distances are evaluated using Great Circle distances. Great Circle distance is an arc measured between two points where the arc is used to establish the shortest path between the two points. Great Circle distance can be easily calculated from geographic coordinates such as latitude and longitude. Great Circle distance (D) between two points is evaluated by the following formula in spherical trigonometry:

$$\mathrm{Cos}\,(D) = (\mathrm{Sin}\ a\ \mathrm{Sin}\ b) + (\mathrm{Cos}\ a\ \mathrm{Cos}\ b\ \mathrm{Cos}\ |c|)$$

Where a and b are represented by latitude coordinates and |c| is the absolute value of the difference of longitude between the respective coordinates. The Teitz and Bart (1968) heuristic, that is used in this thesis, uses the Great Circle distance calculation to compute the distance between demand points and potential facility locations. All demand and facility locations that are used in the present research are modelled using latitude and longitude coordinates. For a more detailed explanation of the Great Circle distance formula, refer to Muehrcke and Muehrcke (1992) Figure 12.3, page 255.

Solutions for LA problems typically portray the facilities and demand nodes as points and use straight lines, known as desire lines, to connect each demand node to the facility it has been allocated to. Since each demand node is only allocated to one facility, the resulting diagram looks similar to a spider's legs and for this reason, the final diagrams have been called spider diagrams. Ideally, the final diagram would also include a graphic representation of the weight associated with each demand node. Each demand node would be represented by a point located at its centroid or the center of the zone that the demand node is located in. The demand weight would be displayed using proportional circles where a demand node having a larger demand weight would be displayed with a larger circle than a node with a smaller demand weight.

Although most LA researchers do not presently deal with the background mapping component, the background or outline of the analyzed area can be represented as a choropleth map. A choropleth map shows the distribution of a phenomenon by graded shading to indicate the density per unit of that phenomenon (Mayhew, 1997, p. 75). Choropleth maps are an excellent method of visualizing the distributions of the variables that are being analyzed.

Much of the research performed in the LA field is essentially done to derive new methods and solution techniques. Bradley (1989) describes the implementation and development of a system to support optimization experts. The system would give the location researcher access to a system which supports model building, optimization, model libraries, high level solution algorithms, methods to share literature and contact colleagues. The article does not mention a venue to test or evaluate these models against real world data sets. It is interesting to note that the researchers did not mention that these models must to be tested against real world data even though the models are developed to model real world situations. Literature relating to the topic of location analysis practitioners using randomly generated data sets versus real world applications is very sparse and reflects the attitudes of the location analysis community. Many in the location community have determined that real world data sets are difficult and time consuming to deal with. As a result, many researchers feel it is more practical to simply produce randomly generated data sets and examples. It is very difficult to find any discussions about randomly produced data sets and their implications in the LA literature.

Rosing and Hodgson (1996) in a review of location applications research, argue "… there exists a certain tension in the field of LA concerning the relative importance of application-oriented studies in location-allocation…" (p. 78). They examined this issue further and concluded that there are basically two groups that can be identified in the LA community. The first group is defined as those who usually "…utilize random or otherwise contrived examples (usually small) to test and to demonstrate aspects of solution techniques or of the problem." (Rosing & Hodgson, 1996, p. 79). As Rosing and Hodgson (1996) point out "…many researchers do not use real world data sets because the process of collecting them is laborious, expensive, and for many of them who are not data experts, beyond their comprehension." (p. 79). The second group defined by Rosing and Hodgson (1996),

> "…is composed of those individuals who use real world data sets to test and formulate models and these models have direct implications to real applications. However, this group is plagued by many difficulties; the data sets they use are very large, the data sets need to be validated in terms of their accuracy, they are difficult to acquire and require large amounts of computing power to find effective solutions. In addition, the location researcher using real world data normally requires highly developed solution techniques to solve their problems optimally." (p. 79).

The problem of using real world data is further complicated as some researchers use surrogate data to represent or stand in for another population. For example, researchers use data for the entire population (total population) to represent a specific group such as the number of people less than five years of age. Again, it is difficult to find any references to this type of condition within the location-allocation literature.

Literature within the LA field is very vast and of all the articles written Rosing and Hodgson (1996) point out that they were able to identify only 200 articles that dealt specifically with LA and real world data sets. The researcher shall not review all of this literature but rather cite a few examples. One such study by Nambiar, Geldus and Van Wassenhove (1989) analyzed the location of a new industrial rubber plant in Malaysia. The article discussed the difficulty in obtaining and building a database with the appropriate variables needed for the study. However, Nambiar et al. brought to light an important aspect of their research that many within the LA field have over looked. They state that after their databases are constructed, they will then form the input for assisting decision-making for their present study and future studies (Nambiar, Geldus & Van Wassenhove, 1989, p. 25).

Hodgson, Rosing and Storrier (1996) use real world data to test the flow capturing LA model and conclude that actual travel demand data were an important aspect for testing their location model development. The study raises an important data issue as the data sets used by Hodgson et al. contains a "geographical" structure due to the layout of the city and its population distribution. However, the researchers concluded that this inherent structure is acceptable because their model is designed specifically to solve problems based on real world situations. Rosing and Hodgson (1996) point out that "the specificity of the example will frequently be seen as limiting to its publishability, even if confidentially allows it" (p. 80). They do however, state that both approaches, which are those who use real world data and those who do not, have their advantages and make unique contributions to LA research. Rosing and Hodgson (1996) also support the notion that an expanded interaction between the two groups would be to their mutual advantage and to the location analysis field as a whole.

Real world data structure

It is well known that the population within a city is not randomly distributed. As was previously discussed, Hodgson et al. (1996) found that real world data contain an inherent structure. This structure is a direct consequence of how cities grow and mature. Urban geographers and other social scientists have studied the distribution of people within North American cities. Several very influential researchers who attributed particular spatial structures to the growth and development of the typical North American City are Burgess (1923), Hoyt (1939) and Harris & Ullman (1945)

A recent study by Nord (1998) looked at the migration of rich and poor populations and the spatial movement of poverty. He noted that poor people are not well mixed with the more affluent rather, the poorer individuals are concentrated in the center of larger cities and in certain rural areas. Davies and Murdie (1991) performed factorial analysis on 24 Census Metropolitan Areas (CMA) in Canada to determine urban social differentiation. Davies and Murdie (1991) found that large centers, of over 1 million people, displayed a more complex social structure. The researchers also noted that all cities are mixtures of general and local characteristics (Davies & Murdie, 1991, p. 77). The GIS/LA system that is created for this research differs from the Davies and Murdie (1991) study because the 1996 Census of Population data are stored in a Relational Database Management System and accessible through a GIS. This means that the data for this study can be quickly queried or searched based on some type of constraint and then the results of the query can be mapped using a GIS. Davies and Murdie (1991) do not mention the use of tools such as a RDBMS or a GIS that helped them fulfill their research objectives.

To this point, the literature review analyzes the reluctance of LA researchers to use real world data to test their models. The next section looks at how some individuals in the location field are utilizing GIS to meet the needs of their LA research.

Geographic Information Systems and model integration

GIS is a computerized system that stores and manipulates data where geographical location is important. GIS evolved from many well-established disciplines such as cartography, database management systems and statistics. The essence of GIS is its ability to transform geographic data into geographical information useful for decision making.

A few individuals in the LA community have started to recognize the potential of GIS and its applications to location models. Carver (1991) examines how well a multi-criteria evaluation can be integrated with a GIS. He concluded "…GIS provides a suitable framework for the application of spatial analysis methods, such as multi-criteria evaluation, which do not have their own data management facilities for capturing, storage, retrieval, editing, transformation and display of spatial data" (Carver, 1991, p. 321).

Chuvieco (1993) looks at the integration of GIS with generally accepted models such as linear programming. He believes that more effort should be taken to adapt these models to a GIS context. He states that GIS provide an excellent method for data gathering and mapping the linear programming results. Chang and Lin (1997) looked at integrating a GIS and an integer programming model to find the optimal siting of solid waste transfer locations for a metropolitan center in Taiwan. The overall goal of their study was to show how a GIS could work with a database management system and a model-based analytical system to establish the siting of transfer stations (Chang & Lin, 1997, p. 2380). The results of their study provide evidence that GIS is very effective, easy to use and is applicable to location modeling. Miller (1996) examines the integration of GIS and location models from the standpoint of geometric representation in facility location problems. He concludes that the geometric manipulation and analytical capabilities of Geographic Information Systems provide an opportunity for increasing the sophistication of client and facility representation in facility location problems (Miller, 1996, p. 812). He concludes that by utilizing a GIS, the users are able to model facilities as lines and polygons which will have the potential to increase the relevance and flexibility of facility location models (Miller, 1996, p. 812). He suggests that through the use of a GIS, LA researchers can now consider points, lines and polygons as possible client and facility objects and potentially reduce errors in measuring client-facility interaction costs.

Miller and Storm (1996) discuss the complexity and difficulty of creating a database for use with a GIS and network equilibrium-based travel demand models. Although the present study is not undertaking as complex a database design, it is interesting to note that Miller and Storm found that the use of a spatial database with a GIS seems to enhance the ability of a GIS to provide efficient spatial data management, user accessibility, and graphic display of their model results (Miller & Storm, 1996, p. 388). Valeo, Baetz and Tsanis (1998) use a GIS to determine the location of several new recycling depots for the town of Dundas, Ontario, using the maximum covering model. They were able to do all of their analysis including the LA component of their study within Arc/Info®, which includes a maximum coverage model. A weakness of the maximum coverage program included with Arc/Info is that it does not offer the user the ability to use more than one random start. This means that the Arc/Info program is not guaranteed to find an optimal solution. With only one start, the final solution may not be the optimal solution as the solution found may have fallen into a local minimum. With more than one random start, the solution has a greater probability of avoiding a suboptimal solution due to a local minimum.

At the present moment, there are essentially two main software applications available for location-allocation analysis, TransCAD® and Arc/Info's Network® extension. Both of these software packages supply the Teitz and Bart (1968) heuristic in addition to a few more location heuristics. Both packages are very useful and effective location tools. However, these products are both very expensive and both have steep learning curves. Due to the limitations of LA models included with some GIS packages, which might perform sub-optimally, many advocate the integration of external LA models with a GIS.

Goodchild, Haining and Wise (1992) advocate one method of integrating models with Geographic Information Systems. The method is what they describe as 'loose coupling' via ASCII files (Goodchild, Haining & Wise, 1992, p. 418). This approach involves using the GIS software for the aspects it is best suited and using other software when needed. This is accomplished by taking the output from the GIS system in ASCII (text) format to be passed to other software programs. They elaborate on this approach by stating that its greatest strength is the logic of performing each task with a software package which is best suited to that task

(Goodchild et al., 1992, p. 419). This approach succeeds in its simplicity, as no new software must be developed or written. However, the drawback to this approach is its dependence on the format of the output from the software package remaining constant and the complexity of dealing with more than one type of software package (Goodchild et al., 1992, p. 419). Chen and Tang (1994) also discuss GIS and model integration and the type of coupling environment. The coupling environment is of course a very important aspect to GIS and model integration, they state that most coupling occurs between the database and GIS and these researchers also advocate the loose coupling approach. The type of coupling environment is important as it dictates how the final software combinations are used. Loose coupling is a very flexible approach for model and software integration and requires little in the way of development time.

Bennett (1997) points out some disadvantages to the loose coupling approach as loosely coupled integration fails to provide: "(1) a consistent interface; (2) consistent data structure; (3) support for the development and modification of models; and (4) user interaction during a simulated event" (p. 337). Yates and Bishop (1998) developed a methodology to follow when integrating existing GIS and modelling systems. They state that the integration of GISs and existing modelling systems is an important activity that enhances the values of each of these systems (Yates & Bishop, 1998, p. 71). Although, some disadvantages exist for the loose coupling approach, the advantages still out weigh the disadvantages from the standpoint of simplicity.

Relation of this Study to the Literature

From the literature review, one notes that little work is presently been done within the LA field to use real world data sets. It is pointed out that presently LA research is being conducted mainly using contrived data sets. Evidence is presented that real world data contains an inherent structure which may affect LA solutions. A few in the LA research community have done some research to integrate Geographic Information Systems and LA models. However, the research has been mostly exploratory rather than developmental to amalgamate the two fields. The goals of this thesis are twofold: first to provide quick and efficient access to real world socio-economic data for LA analysis and socio-economic problem analysis; second, to provide tools that allows a LA researcher the ability to utilize a GIS to collect, organize, manipulate and display Census of Population data for analysis.

Chapter III. Research Problem

Research Problem

The central research problem for this study is how to effectively organize real world data sets and develop GIS tools to utilize the real world data sets in such a manner that it simplifies LA analysis and socio-economic data analysis. To accomplish this, the present study demonstrates techniques to build an effective database warehouse composed of the 1996 Census of Population socio-economic data from Statistics Canada. The resultant database effectively organizes real world data in an efficient manner and this facilitates quick and efficient access to data sets that are useful to LA modelling. The 1996 Census of Population data are also available in a format that can be easily read by a Geographic Information System. This in turn means that the data stored in the database can be displayed, queried and manipulated in a GIS. Before the heart of the research problem is addressed in detail, some introductory GIS concepts must be introduced to gain a better understanding of how the research problem is approached in this study.

GIS Fundamentals

The fundamental difference between a GIS and a Relational Database Management System (RDBMS) is that GIS uses geographical space as an indexing system. This specialized indexing system allows the user to display multiple maps or "layers" on top of one another, similar to a layered cake. This special type of indexing system allows GIS the ability to perform tasks such as georeferencing (data transformation) and spatial joins that cannot be performed in a RDBMS. Haug (1993) discusses the GISs ability to show relationships between different phenomena.

> "The purpose of a GIS is to represent the real world by means of a two-or three-dimensional model. Such a system allows the processing and modelling of the data that it contains on the basis of a homogeneous spatial reference system. The objects of the real world are represented by the description of the situation (geometry) and specific properties (attributes) as well as by the spatial relations (topology) between the objects" (Haug, 1993, p. 12).

A fundamental difference between a RDBMS and GIS is that GIS provides the ability to display and manipulate spatial data in ways that a RDBMS does not offer.

Geographic Information Systems use two types of data subsystems; spatial and non-spatial data. The 1996 Statistics Canada Census of Population data are available in both spatial and non-spatial formats. Consequently, the data are well suited for analysis using GIS. Spatial data are composed of a digital map layer that include a specific locational component and use some type of geographical referencing system. The referencing system that is used in a GIS ranges from latitude and longitude to a simple Euclidean coordinate system. Spatial data can be subdivided into two data models; vector and raster data. Vector data represent the physical characteristics of an object by representing an object as either a point, line or closed area (polygon). The topology (spatial relationships) of features in the vector data model can be clearly identified and they offer a greater precision in computing spatial properties and processing map features over a raster model (Chou, 1996, p. 67).

Typically, for socio-economic GIS applications such as LA modelling, a vector-based GIS is the model of choice. The present study uses vector based spatial data from the 1996 Canadian Census of Population. The reason vector models are more conducive to this type of analysis is because they have the ability to store large quantities of attribute data. This is an essential component of the present study. In addition, vector models are able to graphically display objects better than raster data model. Raster models, on the other hand, do not offer the ability to store large quantities of attribute data and the graphical quality of these models is not as conducive to Census analysis.

There are; however, a few issues that must be addressed when using a GIS for socio-economic analysis. The needs of the study must be addressed before any analysis or development begins. A few considerations to take in to account are; the level of detail that is required for the spatial database, the availability of the data, size of the database(s) and finally the purpose of the study. The data stored in the databases must be made to satisfy the users' demands with respect to the spatial coverage and levels of aggregation (United Nations Report, 1997). Thus, a choice must be made concerning the type of GIS data model used, whether it is raster or vector, and the level of aggregation required for the study. The type of data model and level of aggregation are important issues because they establish the data storage requirements for the study. Raster data typically require a substantially larger amount of hard disk storage space than vector data. Disk storage requirements also apply to the level of aggregation because as the level of aggregation becomes smaller the disk storage requirements increase. After analyzing the needs of the present study, the researcher chose to use the vector based 1996 Census of Population spatial data sets available at the Enumeration Area (EA) level of aggregation. The EA level of data aggregation is the smallest level of data aggregation available for the 1996 Census of Population and it was decided that these data meet the needs of the present study. The EA level of data aggregation is also very suitable to LA modelling because the problems formulated with these data will be manageable and are easily solved in a timely manner.

Non-spatial data are also referred to as attribute data and do not contain any locational information. Instead, attribute data correspond to the characteristics of the object represented by the spatial data. For example, a point in the spatial database could represent a house and that house in turn has many characteristics or attributes such as its age, the number of bedrooms, size and so forth. Typically, an external database management system or some type of spreadsheet software is used to manage attribute data. For the present study, the attribute data from the 1996 Census of Population are modified and then stored in an external database using some commercial available RDBMS software. The ensuing data warehouse forms the "heart" of the study and some GIS tools are developed around the database to fulfill the data requirements for the LA model.

This research demonstrates how a desktop GIS, ArcView® 3.1, can be customized in order to facilitate real world data manipulation, data querying and to display the data stored in the

Census Attribute Database. Currently the location-allocation tools available for the desktop GIS packages, such as ArcView or MapInfo, are limited. ArcView is selected as the GIS software package for this research because it is an industry standard for desktop GIS packages, it is relatively inexpensive, easy to learn, and it is very easy to customize and integrate LA models. The GIS is used to map the results or solutions of the LA program. The location program and the Census Attribute Database are integrated with ArcView using a loose coupling approach to form a GIS/LA system. This is accomplished through ArcView because it is designed with a completely customizable graphical user interface (GUI). As a result, the interface is redesigned to incorporate routines and functions to facilitate the communication between the GIS, the Census Attribute Database and the LA model. The customized ArcView interface provides the LA researcher with tools for data preparation, data visualization, thematic mapping and the ability to map the final solutions of the LA program. Thus, the goal of the study is met through the development of the database and the customized GIS tools to demonstrate that real world data are very accessible and easy to use to analyze LA problems when it is combined with the functionality of a GIS.

Location-Allocation Model

The location-allocation analysis is conducted using the *p*-median model as the LA model. For many problems, there are better models than the *p*-median model; however, for this study the model provides a very good objective function that is very easy to compare solutions. The *p*-median model is "...the classical and most widely used model." (Hodgson & Storrier, 1995, p. 61). However, it is very difficult to solve large *p*-median problems exactly, therefore, to overcome this limitation, the Teitz and Bart (1968) heuristic is used. Although the Teitz and Bart (1968) might not be the most recent heuristic to our knowledge, it is does provide very good solutions.

Location-Allocation Study

After the two goals of this research have been met, a small research study is undertaken to test the applicability of using real world data to test LA models. This aspect of the study is a modification of a study conducted by Hodgson and Bayne (1990) that examined multi-service facilities in Edmonton. The present study looks at the distribution of two demographic groups, those being the children less than six years of age and the elderly people composed of those individuals who are 65 years of age and older. To do this, the customized GIS tools and the 1996 Census of Population data stored in the Census Attribute Database are used. The LA analysis is conducted for fifteen Canadian cities. The study calculates a set number of optimally located facilities for each demographic group (preschool children and elderly). Combined with this, the researcher calculates how disadvantaged the preschool children and elderly population would be, if the facility was located for a different demographic group (surrogate) such as the total population. This portion of the study is a response to a number of LA researchers who are advocating the use of real world data for LA model testing. Consequently, this research demonstrates how easily real world data can be used with a location-allocation study.

Limitations of this research

One limitation of this study is that it does not consider the data aggregation problem, therefore only deals with the smallest level of aggregation for the 1996 Census of Population data. Murray and Gottsegen (1997) looked at the effects that data aggregation has on the stability of the *p*-median location model. Although, data aggregation is beyond the scope of this study, it is worth mentioning that Murray and Gottsegen (1997) found there is a high level of solution stability when aggregated data are used. Hodgson, Shmulevitz, and Körkel (1997) looked at the aggregation effects on the *p*-median model when aggregating from Census Enumeration Areas to Census Tracts in Edmonton. They discuss three sources of data aggregation error and they discovered a fourth source of aggregation error, dubbed *Source D error*. *Source D error* results from the representation of facility sites by discrete points. They stated that *Source D error* can only be eliminated by disaggregation.

A second limitation is that the end user of the customized GIS interface and database are required to purchase the necessary software in order for it to function effectively. In addition, the end user is required to have some type of familiarity with the GIS and RDBMS software. The present study assumes that the end user is using a personal computer running a current version of Microsoft's® Windows® operating system.

A third limitation is a result of using the loose coupling approach to integrate the GIS, database management system and the location model. The loose coupling approach allows other models to be integrated easily; however, the model must have the appropriate data structure in order for the GIS to read the solution files. Therefore, the integration of the GIS/LA system is inflexible in its design.

A fourth limitation is that the present study does not take into account the effect of suburbs on the distribution of the two demographic groups analyzed. This suburban trend may affect the structure of the larger metropolitan city as younger families live in the suburbs thus potentially skewing the results as it is assumed fewer children and parents are living in the main metropolitan center. For example, some Canadian cities such as Edmonton and Victoria are surrounded by several suburban communities all of which have a very young population base. These smaller communities will affect the overall distribution of the children's population and will not be incorporated into the analysis conducted by this research. This suburban trend may not be present for all the fifteen cities analyzed.

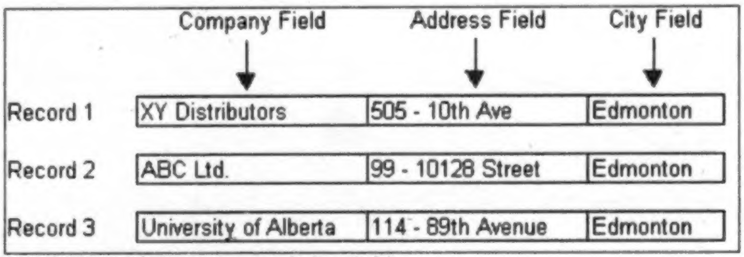
Chapter IV. 1996 Census of Population Data

To meet the first objective of this study, building a data warehouse for quick and efficient access to real world data, one must have some understanding of database fundamentals and the data available for the 1996 Census of Population. In this chapter, the 1996 Census of Population spatial and attribute data are described. Furthermore, this chapter demonstrates techniques to construct a real world database which is integrated with a desktop GIS and a location-allocation model. The Census Attribute Database developed for this research is also a very useful tool on its own to individuals who analyze or study the characteristics of the Canadian Population such as Demographers or Business Analysts. This is because the database allows analysts the ability to quickly retrieve important Census data for any Canadian CMA and use these data for analytical purposes.

Database Fundamentals

A database "... is a unified computer-based collection of data, shared by authorized users, with the capability for controlled definition, access, retrieval, manipulation and presentation to the data within it..." (Worboys, 1997, p. 48). Databases allow for the collection and maintenance of information with an objective of relating or creating an association between information that was previously separate. In simplified terms, a database is composed of a collection of objects called tables, queries, forms and reports which allow the user to control data access and management. Of the many database fundamentals, this study only discusses tables and queries as they relate to the construction of a Census Attribute Database for LA modelling.

Computer scientists have developed several database models for quick and efficient access to data. Most databases developed today are relational in nature and this model is discussed in this study. One of the main components of any database are tables. Tables are objects that store physical data the user wants to manage. The organization of a table can be described in terms of individual records and fields. A record is one row in a table representing information relating to one data entry, as shown in Figure 4-1. In this example, the first record contains information only about the customer called XY Distributors, no information for any other customers are contained in the first record. A field is one column in a table, each containing single item of data. A field identifies where data can be found in the record. In Figure 4-1, the data fields are company name, address of the customer and the City the customer is located in.



	Company Field	Address Field	City Field
Record 1	XY Distributors	505 - 10th Ave	Edmonton
Record 2	ABC Ltd.	99 - 10128 Street	Edmonton
Record 3	University of Alberta	114 - 89th Avenue	Edmonton

Figure 4-1 Organization of data as records in a table.

One or more of the fields in a table may be designated as keys, with the primary purpose of increasing the speed that data are retrieved from a table. All other fields not denoted as key fields are referred to as attribute fields. Keys also increase the efficiency of a database search, known as a query.

One of the main differences between a relational database model and other database models relates to the relational database's flexible structure because there is no hierarchy of the data fields within a record or table. Instead, each field in a relational database table can be used as a key. One restriction placed on tables in most commercially available RBDMS are that the tables must contain no more than 256 fields.

A relational database may contain many tables, all of which may be related or associated together through what is called a relational join, which is a method of linking the database's tables together. A join instructs the database as to how the data in each table are related to the data in another table. For example, a join relates the data in each record of one table to one record in another table. Joins are a subcomponent of a query which is a method of searching the information stored in the database's tables.

Queries or database searches can use one or more tables as long as a join has been established between the respective tables being queried. Without a join, data from multiple tables cannot be searched simultaneously as a join is required to query multiple tables. For example, the bottom section of Figure 4-2 shows the results of a query that is performed between three tables with the aid of two joins connecting these three tables together. In this example, each of the tables has a field that gives each record in the table a unique number, typically this field is referred to as a the common linking field. In order to create a join between tables, a field in each of the tables must be identified and designated as the common linking field. The common linking field in each table must contain similar data. The common linking fields do not need to have the same name, only the same data values, such as numerical values or text characters. For example in Figure 4-2, the values in the customer number field from the customer information table and the customer id field in the transaction information table share the same numeric values. So, the first record from the customer information table can be related to the first record of the transaction information table, because the first record in each table is identified by the numeric value 1. In this example, a table is created called the transaction information table which allows the data in the customer information table to be related (joined) to the sales information table, this is accomplished through the customer identification (ID) and transaction identification (ID) fields in the transaction information table. Figure 4-2 shows the results of a query that asks the database for the following fields: company name and address from the customer information table, the product and price fields from the sales information table. A query of this nature would be valuable to someone, if they are interested in obtaining the product and price that each customer has purchased.

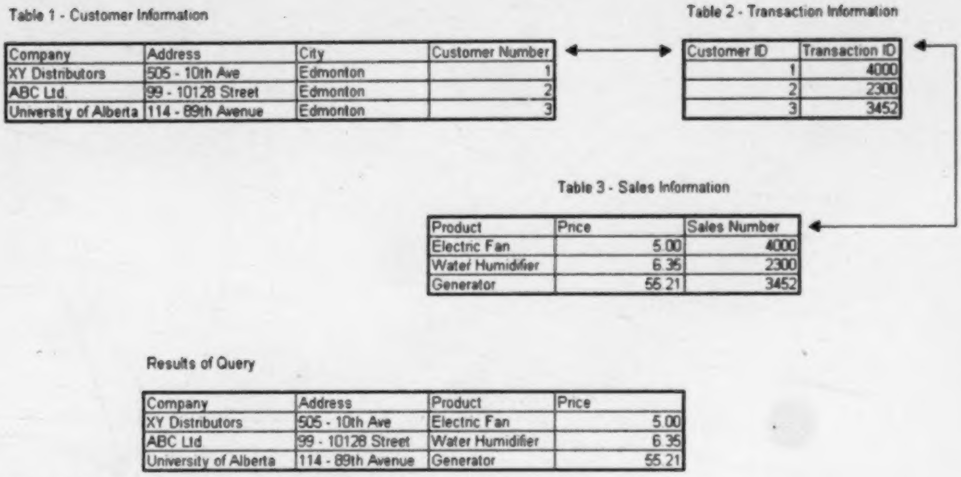


Table 1 - Customer Information

Company	Address	City	Customer Number
XY Distributors	505 - 10th Ave	Edmonton	1
ABC Ltd.	99 - 10128 Street	Edmonton	2
University of Alberta	114 - 89th Avenue	Edmonton	3

Table 2 - Transaction Information

Customer ID	Transaction ID
1	4000
2	2300
3	3452

Table 3 - Sales Information

Product	Price	Sales Number
Electric Fan	5.00	4000
Water Humidifier	6.35	2300
Generator	55.21	3452

Results of Query

Company	Address	Product	Price
XY Distributors	505 - 10th Ave	Electric Fan	5.00
ABC Ltd	99 - 10128 Street	Water Humidifier	6.35
University of Alberta	114 - 89th Avenue	Generator	55.21

Figure 4-2 Relational join and query results.

There are several types of joins that can be used in a relational database. First, a one-to-one join is used where each record in one table is associated to only one record in another table. With this type of join, the common linking fields in both tables have entirely the same values. A one-to-one join is displayed in Figure 4-2 between the customer information table and the transaction information table. Second, a one-to-many join is used where one record in one table is associated with many records in another table. A one-to-many relationship could be used in Figure 4-2 because each customer could potentially have more than one transaction. If this is the case then a one-to-many join is created with one record in the customer table related to many records in the transactions table. In this example, each customer in the customer's table could be related to several transaction records in the sales information table because of the one to many join. Third, a many-to-many join is used where many records in one table are associated to many records in another table and vice versa.

A database's structure can be expressed by a graphic representation, which shows the tables, relationships between tables and field names of each table. A diagram of this type is commonly referred to as an entity relationship diagram. In essence, an entity relationship diagram is a database flow-chart depicting the database's design. Entity relationship diagrams show several of a relational database's fundamental components such as each table, each table's respective fields and the relationships between all of the tables. Entity relationship diagrams depict tables as separate boxes, one to represent each table. Each table's box includes the names of all of the fields contained in it. The fields are described as either key or non-key by their placement in the box. The table's box is separated by a line at the top and any key columns are placed above the line and all non-key fields are listed below. A line is drawn between the tables to represent a relational join. The number 1 and letter M are used to represent the type of relational join used between the tables. The join type identifiers are placed by each table to indicate whether the relationship is a one or a many. The following diagram shows a sample entity relationship diagram for a customer order database.

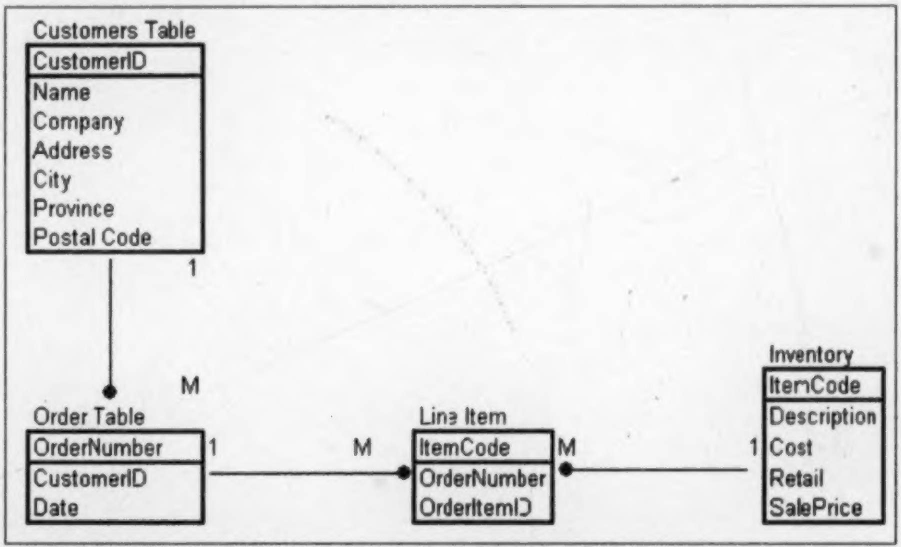

Customers Table
CustomerID
Name
Company
Address
City
Province
Postal Code
1
M
Order Table
OrderNumber
CustomerID
Date
1
M
Line Item
ItemCode
OrderNumber
OrderItemID
M
1
Inventory
ItemCode
Description
Cost
Retail
SalePrice

Figure 4-3 Sample entity relationship diagram for a Customer Order database.

The relational database model has several major advantages. First, the relational database model is more flexible than other database models. The structure of a relational database model does not impede or restrict the type of data processing that can be performed. Second, the organization and structure of a relational database model is easy to understand and replicate. Third, a substantial reason to use a RBDMS is that the database software preserves the join structure created between all of the tables, this means that it does not need to be recreated for each subsequent usage.

Many GIS projects are built around a relational database where the database serves as a data repository for the attribute data to be used in the GIS. The attribute data are then linked or joined to the spatial data sets in the GIS software. A Relational Database Management System (RDBMS) is a sophisticated piece of software that combines all the elements discussed above into one convenient package. The current study uses a RDBMS to create a data warehouse of attribute data from the 1996 Canadian Census of Population and a GIS package is used to handle the spatial database files.

1996 Census of Population Data

The 1996 Census of Population involved over 29 million Canadians as data were collected on every Canadian household on census day, May 14, 1996. Statistics Canada used four types of Census questionnaires (2A, 2B, 2C and 2D) to collect the 1996 Census. The 2A questionnaire was answered by 80% of the Canadian population and is referred to as the short form. The remaining 20% of the total Canadian population filled out the more comprehensive 2B, 2C and 2D questionnaires, known as the long forms. The entire Canadian population answered questions relating to age, sex, marital and common-law status. The more comprehensive long form questionnaires requested information concerned with the respondent's first language learned, home language, place of birth, citizenship, ethnic origin, education background, place of residence, place of work, occupation and income.

Statistics Canada collected the data from the questionnaires and captured it into their internal databases. Questionnaire data are released to the public after they have been thoroughly checked for accuracy and quality. The questionnaire data are stored in what Statistics Canada calls "Nation Tables" and those form the basis for the Census Attribute Database developed in the present study. To supplement the Nation Tables, Statistics Canada has developed spatial database files that define the geographic boundaries of the Census attribute data. The spatial data sets are extremely important as these data facilitate mapping and allow for further analysis using the socio-economic variables from the Census Attribute Database.

The spatial database files for the 1996 Census of Population serve to supplement and support the usage of the attribute data collected from the 1996 Census questionnaires. The Census of Population spatial data are available at ten levels of aggregation: Provincial and Territorial boundaries, Federal Electoral Districts, Census Divisions, Census Subdivisions, Census Metropolitan Areas, Census Agglomerations, Census Tracts, Designated Places, Urban Areas and Enumeration Areas. The present study is concerned with the use of Enumeration Areas at the Census Metropolitan Area level and only discusses their usage.

The smallest available level of Census aggregation is at the Enumeration Area (EA) level. An Enumeration Area, as defined by Statistics Canada, is "the geographic area canvassed by one census representative" (Statistics Canada, 1996, p. 66). Each EA contains between 250 - 400 dwellings. Statistics Canada has grouped EA zones into Census Metropolitan Areas (CMA) where a CMA is defined:

> "A very large urban area (known as the urban core) together with adjacent urban and rural areas (known as urban and rural fringes) that have a high degree of social and economic integration with the urban core. A CMA has an urban core population of at least 100,000, based on a previous census. Once an area becomes a CMA, it is retained as a CMA even if the population of its urban core declines below 100,000. (Statistics Canada, 1996, p. 64).

This study utilizes the available CMA spatial data from the 1996 Census of Population, which cover over half of the entire Canadian population and are used as the location data in the LA model. The present study is able to use the 1996 Statistics Canada data set because of the University of Alberta's site license agreement with Statistics Canada, which allows the University to distribute the data free of charge for research and teaching purposes.

Census of Population - Attribute Data

In order to access the Nation Table data, Statistics Canada recommends "Beyond 20/20," a data browsing software. Beyond 20/20 is an easy to use data browsing software, which allows data to be imported and exported easily. However, a major disadvantage of the Census attribute data and the Beyond 20/20 software is that it is very difficult to use the software to query the data to extract Census data for a particular Canadian city such as Edmonton. For example, the software does not allow the user to simply create a query that would only search for data relating to Edmonton, this problem is directly related to how the data have been created. The Census of Population data are coded and categorized so that queries of this nature are not possible. This presents a major obstacle to the present study and is discussed in greater depth in the next section. A second disadvantage with the Census attribute data relates to its proprietary data storage format because this format cannot be integrated directly with a GIS or a LA model.

Development of a Census of Population Attribute Database

Typically, most GIS applications use a GIS to handle the spatial data and a RDBMS to store the attribute data. The present study is no exception. The present study uses a RDBMS to store the demographic variables from the Nation Tables that correspond to all of the CMA zones available for Canada. The use of the RDBMS facilitates the integration with ArcView GIS and provides real-world data that can be used for LA modelling and socio-economic analysis. In addition, the development of a Census Attribute Database provides quick and efficient access to Census demographic data for any city in Canada that is contained within a CMA region.

ArcView presently does not have the capabilities to store large quantities of attribute data and the data structure they induce. This is because ArcView is unable to embed and store the critical database structure such as the relational join information or query statements. ArcView is not a RDBMS and consequently it cannot store large volumes of attribute data efficiently. However, ArcView does have the ability to read and manipulate attribute data sets. In addition, ArcView is very effective at joining attribute data sets to spatial data sets. ArcView is able to join ASCII files to a spatial database which is of great importance to this study because the solution files from the location-allocation program are in ASCII format. Since ArcView is limited in its database capabilities, a RDBMS is required to create the Census Attribute Database to store the 1996 Census of Population data.

The Nation Tables have been designed by Statistics Canada in such a way that the spatial data integrates in a straightforward manner, as both the spatial and attribute databases share a common field. All of the Nation Tables contain the Enumeration Area unique identifier (EAuid) field followed by the attribute data corresponding to each EA zone. The EAuid field is a unique 8-digit identifier assigned to each EA zone in Canada. When the EAuid field is broken down into its residual components, it is composed of the 2-digit Provincial code, the 3-digit Federal Electoral District code and a 3-digit code that makes every zone unique. Since the Nation Tables and all of the spatial databases contain the EAuid field, the Nation Tables can be joined together through a relational join thus enhancing the importance of both data types.

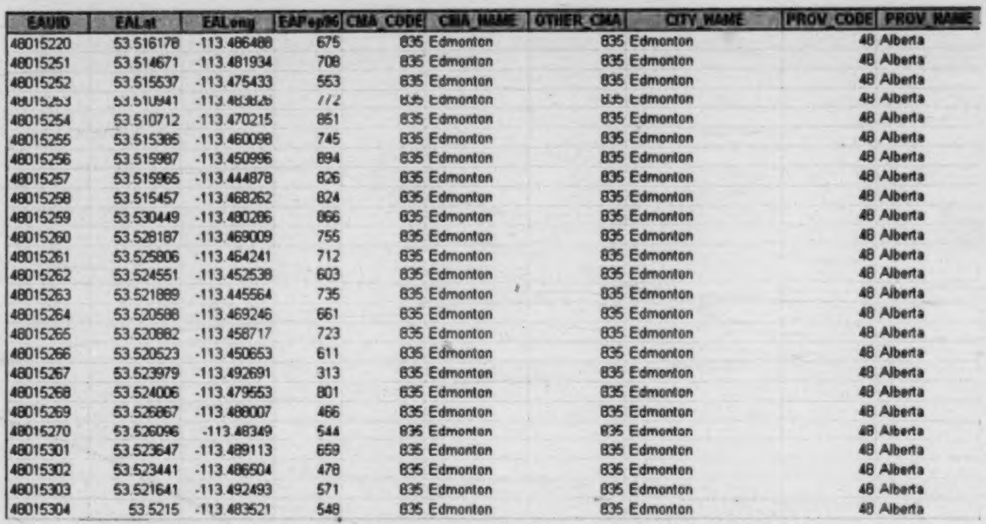
To allow attribute data queries that apply to a particular city, the present study creates an EAuid Lookup table. The EAuid lookup table allows the database to be designed so that the demographic data for any city can quickly be retrieved and integrated easily with a LA model. The EAuid Lookup table is created by taking all of the EAuid numbers for all CMA regions in Canada and then coding the EAuid number to correspond to the city that contains the zone. This allows the EAuid values to be queried based on the city name rather than just a value or range of values. It would be very difficult to extract data for a particular city only using EAuid values because the numbering scheme used by Statistics Canada is not sequential. Instead, they have grouped zones into blocks and have given each block a different numbering scheme. For example, 40 zones in Northeast Edmonton might have values that range from 1 to 40 while the 40 zones adjacent to these zones might have values that range from 780 to 820. Statistics Canada has numbered EA zones in this manner to ensure that they can properly handle the event that an EA zone must be split due to population growth in the subsequent Census. This complicates matters for the end data user and makes it more difficult to code the EAuid lookup table, thus creating an interesting database development challenge.

The GeoRef software issued by Statistics Canada is a tremendous resource for the creation of an EAuid lookup table. GeoRef is a search tool based on the 1996 Census of Population Geographic database used by Statistics Canada which includes information on the 1996 Population, dwelling counts, land area, geographic codes, and in some places, the 1991 Census Population. The GeoRef software is able to produce reports for each CMA region, which lists the Federal Electoral District (FED) and EA code for each EA in the CMA. These reports also indicate the Census Subdivision (CSD) type for all zones. The CSD and FED codes are important to the present study as these data are used to create the very important EAuid lookup table. Each zone is classified by a certain CSD type ranging from a borough to a city.

All of the EAuid values for the 50 CMAs are entered into a database table and each EAuid zone is coded with the following information: CMA 3-digit code, the CMA (primary city) name, name of any secondary CMAs present within the main CMA, the 2-digit Provincial code and name and if a zone is designated as a city, the city's name is entered into the database. The Provincial code and Provincial name are entered into the EAuid lookup table because some provinces have cities with the same names.

There are two extremely important fields in this table. The first is the EAuid, which allows the data to be joined to the spatial databases and the Nation Tables. The second is city name (City_Name). This field allows a query to be designed that will search for all data for a particular city within any of the CMAs in Canada.

The EAuid lookup table, although time consuming to develop is instrumental in this study because the Nation Tables are joined to it in the RDBMS. Because the EAuid lookup table deals only with EAs contained within the CMAs, it contains only 27,671 EA zones of the total 49,502 EA zones that define the entire country. The nine Nation Tables all contain 49,502 EA zones.



EAUID	EALat	EALong	EAPop96	CMA_CODE	CMA_NAME	OTHER_CMA	CITY_NAME	PROV_CODE	PROV_NAME
48015220	53.516178	-113.486488	675	835	Edmonton	835	Edmonton	48	Alberta
48015251	53.514671	-113.481934	708	835	Edmonton	835	Edmonton	48	Alberta
48015252	53.515537	-113.475433	553	835	Edmonton	835	Edmonton	48	Alberta
48015253	53.510941	-113.483826	772	835	Edmonton	835	Edmonton	48	Alberta
48015254	53.510712	-113.470215	851	835	Edmonton	835	Edmonton	48	Alberta
48015255	53.515385	-113.460098	745	835	Edmonton	835	Edmonton	48	Alberta
48015256	53.515987	-113.450996	894	835	Edmonton	835	Edmonton	48	Alberta
48015257	53.515965	-113.444878	826	835	Edmonton	835	Edmonton	48	Alberta
48015258	53.515457	-113.468262	824	835	Edmonton	835	Edmonton	48	Alberta
48015259	53.530449	-113.480286	866	835	Edmonton	835	Edmonton	48	Alberta
48015260	53.528187	-113.469009	755	835	Edmonton	835	Edmonton	48	Alberta
48015261	53.525806	-113.464241	712	835	Edmonton	835	Edmonton	48	Alberta
48015262	53.524551	-113.452538	603	835	Edmonton	835	Edmonton	48	Alberta
48015263	53.521889	-113.445564	735	835	Edmonton	835	Edmonton	48	Alberta
48015264	53.520588	-113.469246	661	835	Edmonton	835	Edmonton	48	Alberta
48015265	53.520882	-113.458717	723	835	Edmonton	835	Edmonton	48	Alberta
48015266	53.520523	-113.450653	611	835	Edmonton	835	Edmonton	48	Alberta
48015267	53.523979	-113.492691	313	835	Edmonton	835	Edmonton	48	Alberta
48015268	53.524006	-113.479553	801	835	Edmonton	835	Edmonton	48	Alberta
48015269	53.526867	-113.488007	466	835	Edmonton	835	Edmonton	48	Alberta
48015270	53.526096	-113.48349	544	835	Edmonton	835	Edmonton	48	Alberta
48015301	53.523647	-113.489113	659	835	Edmonton	835	Edmonton	48	Alberta
48015302	53.523441	-113.486504	478	835	Edmonton	835	Edmonton	48	Alberta
48015303	53.521641	-113.492493	571	835	Edmonton	835	Edmonton	48	Alberta
48015304	53.5215	-113.483521	548	835	Edmonton	835	Edmonton	48	Alberta

Figure 4-4 EAuid lookup table - screen capture of a small section of the table.

During the development of the EAuid lookup table, special consideration was taken with respect to the EAuid field to ensure that it was formatted as a string in the EAuid lookup table. The formatting of the EAuid field is essential as the formatting of each joining field must be identical to allow a join to be performed between any of the Nation Tables and the spatial databases. For the process to work correctly the two tables must have the same values and the values must be in the same type of formatting. The join process involved between the attribute and spatial database files is explained in more detail in the next chapter.

The first step required to create the Census Attribute Database after the development of the EAuid lookup table is to export all of the data stored in the Nation Tables from the Beyond 20/20 data browser. The data must be exported because of the proprietary data storage format that the Nation Tables are stored in. The reason a query is not performed using the Beyond 20/20 data browser to extract data for individual cities relates to the difficulty of extracting data for only certain cities. The data in the Beyond 20/20 data browser are not formatted so that individual cities can be extracted by name instead the data can only be queried based on their EAuid values.

The data from all of the Nation Tables are exported from Beyond 20/20 in database file format (dbf) and are imported into a RDBMS. Each of the Census variables are labeled in the

Beyond 20/20 data browser using a textual description of the variable these data represent. For example, one variable in Nation Table 1 is listed as the "Average number of never-married sons and/or daughters at home per census family". This textual description in some cases may exceed 100 text characters. Since most commercial RDBMS software packages have limitations on the number of characters that can be used to define a field's title, some type of coding scheme is required. Consequently, a 4-digit code is developed for the present study to organize the fields contained in all of the nine Nation Tables and to ensure that each field is uniquely identified. The first digit of the database code refers to the Nation Series table that the data were originally stored in. The remaining three digits represent the location that the field was in the Beyond 20/20® browser. The last three digits are designed to give each category a unique number. A unique number is important so that no field has a duplicate value, thus making each field unique and easily identifiable. Appendix 1 contains the word descriptions on what each database field number refers to in the 1996 Census of Population survey. The following Table is an account of what variables are stored in each Nation Table and how many variables are present in each table.

Nation Table	Demographic variables covered	Number of variables
1	Age and Sex, Marital Status/Common Law, Families (Part I: Number, Type and Structure), and Structural Type of Dwelling and Household size	118
2	Immigration and Citizenship	111
3	Mother Tongue, Home Language	239
4	Aboriginal	8
5	Ethnic Origin, Visible Minorities	316
6	Labour Force Activities, Occupation and Class of Worker, Household Activities, Place of Work, Mode of Transportation to Work, Commuting Distance	539
7	Education, Mobility and Migration	60
8	Sources of Income	165
9	Family and Household Income, Families (Part II: Social and Economic Data), Occupied Private Dwellings, Household and Housing Costs	47

Table 4-1 Demographic variables covered by each Nation Table.

Several of the Nation Tables require special attention as the tables contain more fields than the conventional RDBMS can handle in one table. Most commercial RDBMS can only handle 256 fields in any table. Therefore, two of the Nation Tables are divided into several smaller tables and then imported into the RDBMS. Nation Table 5 is divided into two sub-components and Nation Table 6 is broken down into three sub-components. The division of these two tables into smaller components does not affect data quality or the database design. The reason for this is a direct result of the relational database model. The model allows data in one or several tables to be joined together and data from both tables can be queried as though the data exists in only one table.

The Nation Tables were exported from the Beyond 20/20 browser for all EA zones in Canada and the EAuid lookup table only corresponds to a small segment of the total number of EA zones in Canada. Thus, the Nation Tables contain data that do not correspond to the EAuid lookup table. The data in the nine Nation Tables contains data for each EA zone in Canada (49,502 zones) and the EA Lookup table only contains 27,671 zones, where each of these zones are contained in a Census Metropolitan Area. For that reason, the researcher decided to only extract the EA zones from the Nation Tables that correspond to data stored in the EA Lookup Table. To do this, a temporary database is created where the nine Nation Tables are joined to the EAuid Lookup Table and each Nation Table is individually queried for only those values that are contained in the EAuid Lookup Table. Each table is queried to only return the EA zones that corresponded to the values stored in the EAuid lookup table. By querying the Nation Tables in this manner, each Nation Table is reduced from 49,502 EA zones to 27,671 zones.

The queried Nation Tables are then exported out of the temporary database in the standard database file format (dbf) and then the Census Attribute Database is designed around these smaller Nation Tables and the EAuid Lookup table. A note of caution, Microsoft Access modifies numerical field titles when queries are exported out of Microsoft Access in database file format. Microsoft Access by default changes the numerical field titles in the dbf file. For example, Microsoft Access will drop the first numerical value and replace it with a "Z" character. Next, all the tables are imported into the RDBMS to create the Census Attribute Database, where the tables must be joined together. The Census Attribute Database is created with a structure that uses a one-to-one relational join between the smaller versions of the Nation Tables and the EAuid lookup table. The Census Attribute Database is designed so that the Nation Tables and the EAuid lookup table contain the same number of records. This is only possible because the Nation Tables were queried to only contain the same records that are contained in the EAuid Lookup table. Since all of the tables contain the EAuid field, the EAuid field is a common field shared amongst all of the tables and is used to join the tables together. The EAuid field exists in all of the Tables and is designated as a primary key. The EAuid field is also indexed to enhance the database's retrieval speed. The following Figure shows an entity relationship diagram for the final database design showing a one-to-one relational join. The Nation Tables are all joined to the EAuid Lookup Table using a one to one join relationship.

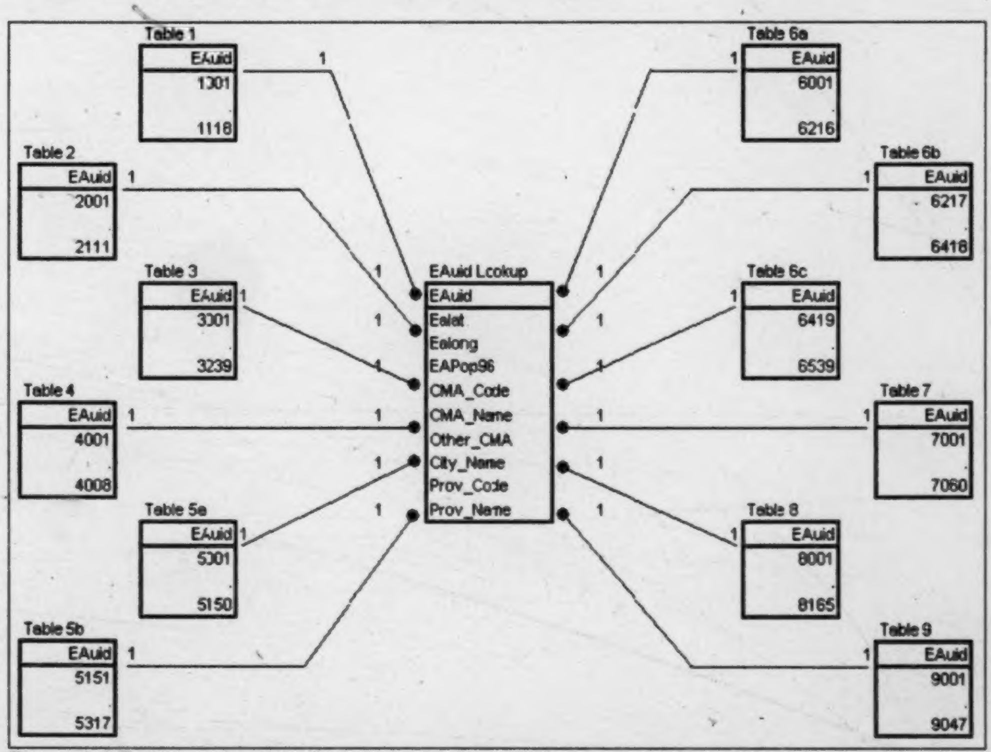


Table 1
EAuid
1301
1118
Table 2
EAuid
2001
2111
Table 3
EAuid
3301
3239
Table 4
EAuid
4001
4008
Table 5a
EAuid
5301
5150
Table 5b
EAuid
5151
5317
EAuid Lookup
EAuid
Ealat
Ealong
EAPop96
CMA_Code
CMA_Name
Other_CMA
City_Name
Prov_Code
Prov_Name
Table 6a
EAuid
6001
6216
Table 6b
EAuid
6217
6418
Table 6c
EAuid
6419
6539
Table 7
EAuid
7001
7060
Table 8
EAuid
8001
8165
Table 9
EAuid
9001
9047
1

Figure 4-5 Entity relationship diagram for the final 1996 Census Attribute Database design.

1996 Spatial Centroid Data

Location-allocation models use aggregated data to simplify the complexity of a problem and make the problems more manageable and easier to solve. The 1996 Census of Population data contains a spatial database that is aggregated to an EA zone's centroid or center point (refer to Figure 4-6). Statistics Canada defines a zone's central point or centroid as a representative point (Statistics Canada, 1996, p. 73). The purpose of the representative points is to assign data to that point for data aggregation, data retrieval, data analysis and statistical mapping such as LA modeling (Statistics Canada, 1996, p. 73). Each EA zone's representative point is identified by a pair of latitude and longitude coordinates stored in decimal degrees. Statistics Canada maintains the values for all representative points in the Geographic Attribute Database and these data are available to the public through the GeoRef software. The representative points data are modelled as points in a GIS.

Part of the intention of this research is to simplify LA modelling; therefore, the researcher decided to integrate the representative point data into the EAuid lookup table. The entity relationship diagram in Figure 4-6 schematically shows that the representative point data are included in the EAuid lookup table. The fields that define the latitude and longitude coordinates are the EALat and EALong fields. A third field, EAPop96, is also included in the EAuid lookup table and is the total population in each EA zone. These three fields are included to simplify the mapping process for the LA modelling with a GIS. One of the main reasons the representative point data are included in the attribute database is to ensure the integration between the Census Attribute Database and the desktop GIS occurs smoothly. The integration between the Census Attribute Database and the GIS software is discussed in more detail in the following chapter.

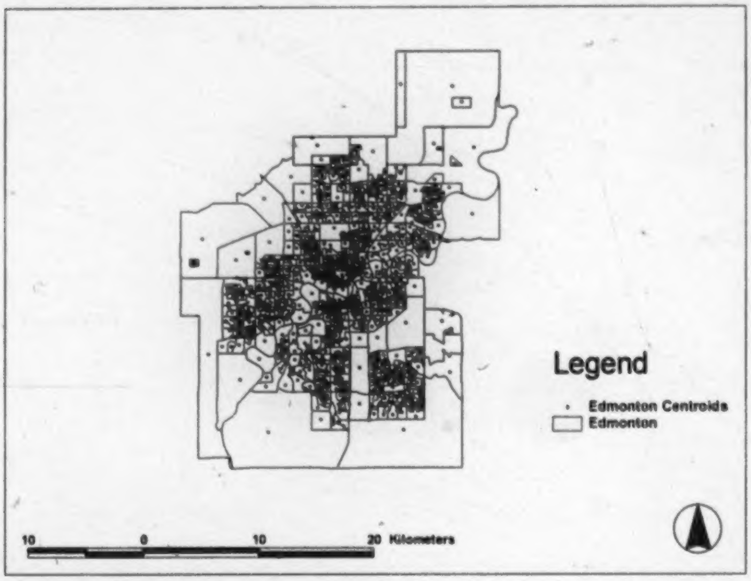


Figure 4-6 City of Edmonton - EA zones with representative points (centroids).

Census of Population - Spatial Databases

The spatial database files for the 1996 Census of Population serve to enhance and support the usage of the attribute data collected from the 1996 Census questionnaires. Within the ten levels of spatial data aggregation, two distinct types of files are available; they are called the Digital Boundary Files and the Digital Cartographic Files. The Digital Boundary Files are more concerned with the official Census boundaries; consequently, they do not portray each EA zone's boundaries correctly when they encounter water or other natural barriers. The Digital Cartographic Files on the other hand, show the true cartographic boundaries and include islands and shorelines when a zone encounters water (refer to Figure 4-7). This is an important aspect to the Digital Cartographic Files as one EA zone might have more than one polygon representing it in the spatial database file. For example, the city of Toronto is represented at the EA level by 1188 EA zones, but these zones are represented by 1200 GIS polygons. The present study uses the Digital Cartographic Files for all analysis performed. One of the main reasons the Digital

Cartographic Files are chosen for this study are due to their cartographic representation and also because accurate area calculations can be made from these files. The 1996 Census of Population spatial database files model all EA zones as polygons, which are closed areas.

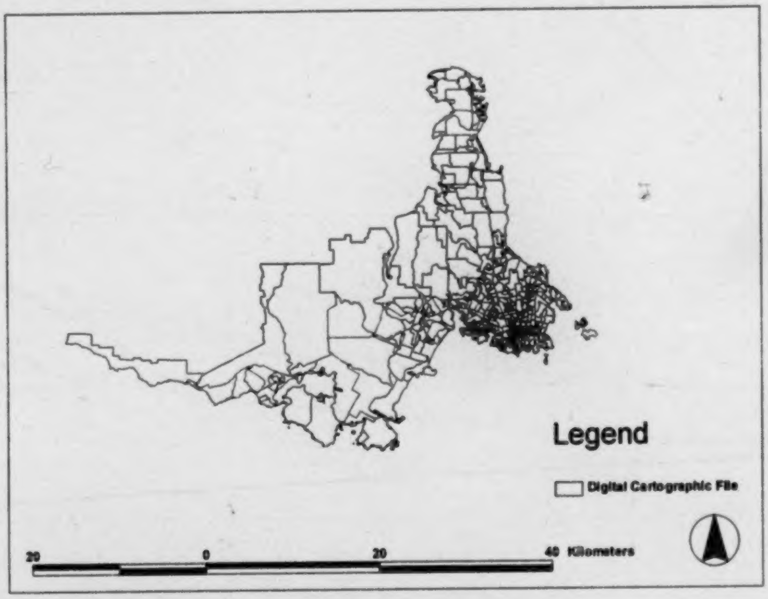


Figure 4-7 Digital Cartographic File for Victoria, British Columbia at the EA level.

The following Table outlines the Digital Cartographic Name for the 26 CMAs that meet Statistics Canada's definition of a CMA. The table also includes the population counts for the major city that resides in the CMA region listed as well as the number EA zone that define the major city.

Census Metropolitan Area (listed by the major city present in CMA file)	1996 Population (thousands)	Number of EA zones per municipality	Statistics Canada Digital Cartographic File Name
Toronto (Ontario)	4,444.70	1200	GEA_535B.E00
Montréal (Quebec)	3,359.00	1735	GEA_462B.E00
Vancouver (British Columbia)	1,891.40	862	GEA_933B.E00
Ottawa–Hull (Ontario–Quebec)	1,030.50	534 - 107	GEA_505B.E00
Edmonton (Alberta)	891.5	826	GEA_835B.E00
Calgary (Alberta)	851.6	942	GEA_825B.E00
Quebéc (Quebec)	697.6	313	GEA_421B.E00
Winnipeg (Manitoba)	676.7	862	GEA_602B.E00
Hamilton (Ontario)	650.4	441	GEA_537B.E00
London (Ontario)	416.1	502	GEA_555B.E00
Kitchener (Ontario)	403.3	238	GEA_541B.E00
St. Catharines–Niagara (Ontario)	389.7	180 - 100	GEA_539B.E00
Halifax (Nova Scotia)	346.8	206	GEA_205B.E00
Victoria (British Columbia)	313.4	169	GEA_935B.E00
Windsor (Ontario)	291.7	271	GEA_559B.E00
Oshawa (Ontario)	280.9	161	GEA_532B.E00
Saskatoon (Saskatchewan)	222.1	266	GEA_725B.E00
Regina (Saskatchewan)	199.2	237	GEA_702B.E00
St. John's (Newfoundland)	177.8	156	GEA_001B.E00
Chicoutimi–Jonquière (Quebec)	166.6	219	GEA_408B.E00
Sudbury (Ontario)	166.2	147	GEA_580B.E00
Sherbrooke (Quebec)	150.0	125	GEA_433B.E00
Trois-Rivières (Quebec)	143.6	88	GEA_442B.E00
Thunder Bay (Ontario)	131.3	176	GEA_595B.E00
Saint John (New Brunswick)	129.1	123	GEA_310B.E00
Abbotsford (British Columbia)	105.4	150	GEA_932B.E00

Table 4-2 Primary CMA regions with their respective population and EA zone values.

Statistics Canada also created CMA files for several cities that do not meet their definition of 100,000 people residing in the city. These files are outlined in the following Table, which lists the 18 CMAs that are below a population count of 100,000 and above 30,000 but have a CMA file.

Census Metropolitan Area (listed by the major city present)	1996 Population (thousands)	Number of EA zones per municipality	Statistics Canada Digital Cartographic File Name
Guelph (Ontario)	95.8	139	GEA_550B.E00
Sudbury (Ontario)	92.1	147	GEA_580B.E00
Kelowna (British Columbia)	89.4	131	GEA_915B.E00
Brantford (Ontario)	84.8	111	GEA_543B.E00
Sault Ste. Marie (Ontario)	80.1	110	GEA_590B.E00
Barrie (Ontario)	79.2	115	GEA_568B.E00
Kamloops (British Columbia)	76.4	107	GEA_925B.E00
Prince George (British Columbia)	75.2	94	GEA_970B.E00
Sarnia (Ontario)	72.8	103	GEA_562B.E00
Nanaimo (British Columbia)	70.1	141	GEA_938B.E00
Peterborough (Ontario)	69.5	101	GEA_529B.E00
Lethbridge (Alberta)	63.1	90	GEA_810B.E00
Red Deer (Alberta)	60.1	79	GEA_830B.E00
Moncton (New Brunswick)	59.3	83	GEA_305B.E00
Kingston (Ontario)	55.9	124	GEA_521B.E00
North Bay (Ontario)	54.3	79	GEA_575B.E00
Belleville (Ontario)	37.1	59	GEA_522B.E00
Saint-Jean-sur-Richelieu (Quebec)	36.4	79	GEA_459B.E00

Table 4-3 Secondary CMA regions with their respective population and EA zone values.

The information presented in these two Tables is extremely important to LA modelling as the number of EA zones defines the size of the problem in terms of how many demand and facility nodes are available. Appendix 2 contains a more detailed breakdown of the CMAs in the 1996 Census of Population and how many EA zones are present for the each CMA as well as for all other cities within that particular CMA. For example, the CMA file for Vancouver also contains the following metropolitan areas: Burnaby, Coquitlam, Langley, New Westminster, North Vancouver, Port Coquitlam, Port Moody, Richmond, Surrey and White Rock. The following Figure shows the Vancouver CMA region and the other ten metropolitan areas contained within the Vancouver CMA.

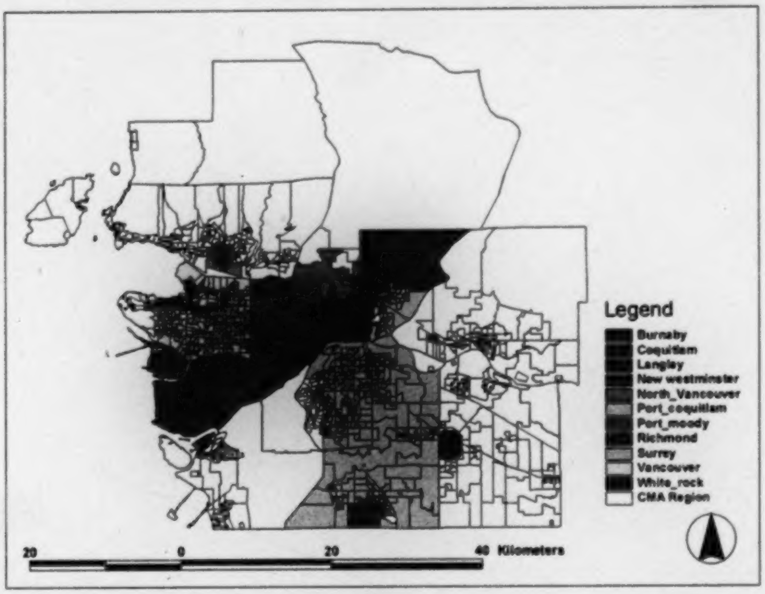

Legend
Burnaby
Coquitlam
Langley
New westminster
North_Vancouver
Port_coquitlam
Port_moody
Richmond
Surrey
Vancouver
White_rock
CMA Region
20
0
20
40 Kilometers

Figure 4-8 CMA region for Vancouver, British Columbia at the EA level. The figure shows how more than one City is contained within one CMA file. In this example, there are ten other cities in the Vancouver CMA region.

The next section describes the Census attribute data in detail from the standpoint of data quality issues that the end user must be aware of before implementing any location-allocation analysis.

Census of Population - Attribute Data Quality, Sampling, Weighting and Random Rounding

Data on age, sex, marital status, common-law status and mother tongue are collected from all Canadians. However, a majority of the Census data collected for the 1996 Census comes from a 20% sample of the population that filled out the long Census questionnaire. As a result, several data issues arise such as confidentiality, area suppression, data rounding and data weighting due to how the Census data were collected.

The Statistics Act of Canada guarantees that a respondent's information remains confidential and is never released to the public in a form that may identify individual respondents. Therefore, the Nation Tables do not record any respondents' names, addresses or telephone numbers. To guarantee confidentiality, Statistics Canada has adopted area suppression and data rounding techniques, where area suppression is used to delete all information for geographic areas with populations below a specified size. In other words, Statistics Canada enters zero values for data suppressed areas. Data suppression applies to any Census areas with a population of less than 40 persons. In such instances, only the total population count is released in the Nation Tables. The 1996 Census of Population income data are suppressed when the population falls below 250 persons in a Census zone. If the quality of the data is determined to be poor by Statistics Canada standards, they may decide to suppress the data in a Census area.

To maintain and to ensure complete confidentiality, the data values in the Nation Tables are randomly rounded up or down to a multiple of 5 or 10 and data values less than 10 are rounded to either a 0 or a 10. This is important to remember when using the data from the Census Attribute Database developed for this study. Consequently, if the data in the Census Attribute Database are summed or grouped, the total value calculated may not match the individual values entered by Statistics Canada for the same EA zone. Likewise, the calculated demographic variables in percentages rely on rounded data and therefore, may not add up to 100%. The only demographic values that are not rounded are the total population counts (stored as variable 1001 in the Census Attribute Database), as they provide no information about the respondent's characteristics.

Census of Population - Spatial Data Quality

Statistics Canada categorizes spatial data quality into five major components: lineage, positional accuracy, attribute accuracy, logical consistency and completeness. Not all of these data quality issues are applicable to this research and consequently they are not discussed.

Lineage includes "...descriptions of the source material from which the data were derived and the methods of derivation, including the data of the source material and all transformations involved in producing the final digital files..." (Statistics Canada, 1996, p. 8). The base maps ranged in scale from 1:2,400 to 1:250,000. All geoprocessing performed by Statistics Canada was performed using Arc/Info version 6.1.1 using a Lambert Conic Conformal projection. The projection for final files was transformed from Lambert Conic Conformal to latitude and longitude using Arc/Info's projection transformation procedures.

The spatial database files originally used for the current study are available in Arc/Info Export format and are converted into a format that is easily read by ArcView GIS. As previously mentioned, the spatial database files rely on the latitude and longitude coordinate system. All of the Census spatial database files are automatically georeferenced because they all use the same coordinate system, latitude and longitude and the Lambert Conformal Conic projection. Georeferencing is the process of ensuring that all spatial database files share the same coordinate system. The spatial database files discussed above are displayed and manipulated using ArcView and are discussed in the next chapter.

CMA Spatial Databases - Metadata

An important aspect to all spatial databases is their associated metadata which is the descriptive information relating to what each piece of data in the spatial attribute database refers to. The metadata for the 1996 Census of Population spatial database files are found in Appendix 3.

Summary

This chapter describes the 1996 Canadian Census survey and the associated data. The spatial and attribute data described in this chapter are data sets for use with LA models. These data require some manipulation to prepare them for use with LA models, but once the data are set up, the data can be continually re-used and manipulated for many types of LA models. This chapter also demonstrates that the spatial and attribute data sets can be used to extract data for individual Canadian cities and then used with a LA model. The information presented in this chapter lays the foundation for the location-analysis. The data contained in the Census Attribute Database are used as the demand weight data in the LA model and the spatial database files supply the location coordinate information to the LA model.

Chapter V. Integrating GIS and Location-Allocation Models

The second major objective of this study is to integrate a desktop GIS and LA model to create an integrated GIS/LA system. A loose coupling approach is used to integrate the GIS software, the LA model and the Census Attribute Database. The basic philosophy of the loose coupling approach is that each of these software packages works in concert with the others to simplify LA modelling. The reason this approach is taken is because it allows the GIS/LA system to use the strengths of the GIS, the LA model and the RDBMS software. A substantial advantage to the loose coupling approach is that it does not require a lot of software development time nor programming knowledge. The present study integrates and modifies the desktop GIS software, ArcView, around the requirements of the LA program. The Census Attribute Database, discussed in the previous chapter, is also integrated with the GIS software so that the data are used with the LA program. The integration is accomplished through the development of tools for the GIS software, where the tools aid the user in the creation of the necessary data files for use with the LA program. The GIS tools are designed to simplify the mapping of the results computed by the location model. Furthermore, the GIS tools include some other functions that allow a location researcher to quickly create cartographic background images of the city being analyzed.

This chapter is organized in major sections. The first introduces ArcView through a very broad and generalized overview. The second section discusses in conceptual terms the integration approach that is undertaken by this study to integrate ArcView, the LA model and the Census Attribute Database. The third section discusses the specifics of the input and solution files that are used and created by the LA model. The fourth section discusses the basic methodology describing how the ARCLA extension is designed to manipulate, organize, maintain and visualize geographic data. The final section discusses some of the extra utilities that are embedded in the ARCLA extension.

ArcView GIS Overview

ArcView is divided into ArcView documents, which are organized in a hierarchical structure. The main document, known as the project document is a working file that organizes all of the other ArcView documents, which are the views document, tables document, charts document, layouts document and scripts document. The top level or main display window in ArcView is the project window. The project window organizes all the files and all other documents that the user generates while using ArcView. When an ArcView project is saved and reopened, the project is returned to the exact state of the last save, with all windows, associated graphics, and tables displayed as before. A project file is dynamic, meaning that the project file stores the steps that are performed rather than the physical data.

ArcView projects contain a listing of all the available ArcView documents, which are represented as icons in the project window. Each document is comprised of its own menus, functions, buttons and tool bars. This is an important factor to consider when using ArcView because the user must be aware that each document contains different menus, buttons, tool bars and functions.

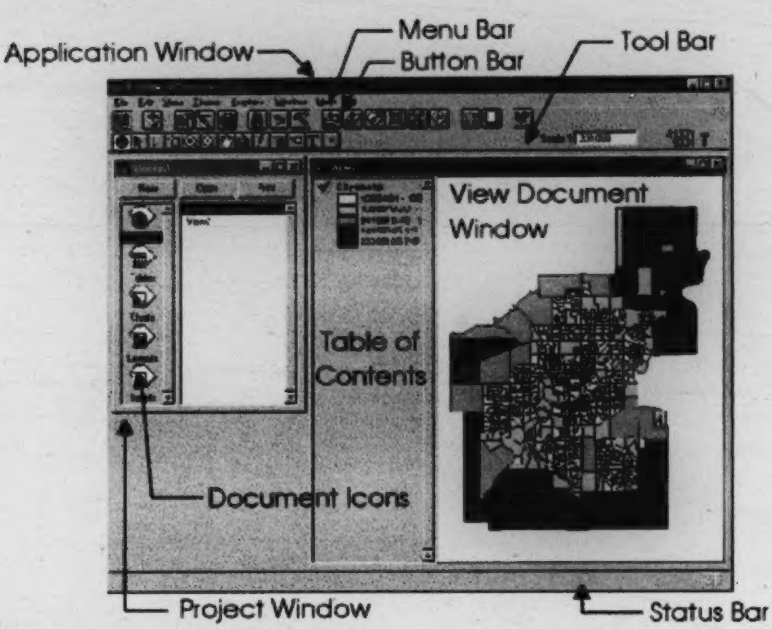

Application Window
Menu Bar
Button Bar
Tool Bar
View Document Window
Table of Contents
Document Icons
Project Window
Status Bar

Figure 5-1 ArcView's Graphical User Interface.

ArcView Views

View documents organize and display the graphical component of spatial database files. In Figure 5-1, the view document is displayed with the Census spatial data file for Edmonton in the view window. Views allow the user to visualize, manipulate and query spatial data sets. Spatial database files in ArcView are called themes or shapefiles. The view window is composed of two components, on the left is a list of all the themes in that particular view and this area is referred to as the table of contents. One or more themes are considered to be in an "active" state if the theme appears to be highlighted or raised in the table of contents. At least one theme must be set to an active state to perform any view functions or operations within ArcView. The right side of the view window is the graphical display area where the themes are drawn.

Themes in ArcView come in three different spatial data types: points, lines and polygons. The present study uses point and polygon data, because the 1996 Census of Population spatial database files are available in this format.

Each ArcView theme is composed of a graphical component and a table known as an attribute table where the data in the attribute table directly correspond to the graphical component. The attribute table stores all of the attributes that are associated to the points, lines or polygons that are represented in the graphical component. The attribute table is linked to the graphical component using a one-to-one join relationship between the theme and its table.

ArcView possesses a spatial query function that allows a spatial data file and its associated attribute table to be queried simultaneously. This means that a theme can be queried from the view window and the query is reflected in the attribute table associated to the theme. The query process also works when a query is performed on a theme's attribute table and results are displayed in both the attribute table and the theme. The manner in which a map can be queried and the resulting query is reflected in both the spatial data file and the attribute database is what sets a GIS apart from a conventional.

Each theme contains a legend, which displays information about the theme's features and is represented using some type of symbology. Themes can be modified by changing the legend through the legend editor dialog box, which allows the user to change the type of symbology, colours and patterns displayed. The legend editor also allows the user the ability to change which database field is displayed and the number of classes the corresponding data is displayed in. Symbols are assigned to each class so that the distribution of each individual class can be displayed on a map for analysis and presentation purposes.

ArcView Tables

The database capabilities within ArcView are available in the table document. Tables or databases are added to ArcView in three standard formats: text files (txt), database files (dbf) and Arc/Info table format. The table document organizes an ArcView project's databases, theme attribute databases and any stand-alone tables. The table document allows the user to query the data sets, add or manipulate data, export data and perform joins between tables. Queries are performed using a modified SQL syntax created for ArcView. The query window displays a selection of fields, operators and values from a table that the user can use to construct a database query. After a query is made and executed, the selected elements are highlighted in the table and on the view. ArcView provides the ability to join tables based on common linking fields that are shared between the two tables. ArcView's join function is robust as it allows tables of different formats to be joined and stored in what is referred to as a virtual table. For example, ArcView provides the ability to join a text file containing tabular data to an ArcView theme's attribute table. This functionality is important to the current research study. The resultant table is referred to as a virtual table as the information is not actually stored in a physical file rather the data from the two tables are simply linked together without requiring extra hard drive disk storage space.

ArcView Charts

The chart document is used to graph or chart the data that are used in ArcView. The charting capabilities in ArcView are insufficient for most applications, because the chart document cannot graph large and complex data sets. In addition, the types of charts that can be created are also limited. Therefore, it is highly recommended that any graphing be performed using software such as Microsoft's Excel®.

ArcView® Layouts

Layout documents are useful for combining the above documents (views, tables and charts) into a single document for printing or presentation purposes. Layouts are extremely good for creating cartographically correct presentation images because layouts allow the user to add cartographic elements such as; north arrows, scale bars and text.

ArcView Scripts

The final type of document is called the scripts document. The scripts document uses an object oriented macro language called Avenue® that allows the user to customize and enhance the functionality of ArcView. Avenue is a compiled language that is broken down into a set number of hierarchical classes and functions that the user uses to automate certain tasks to add more functionality to ArcView. ArcView's interface is completely customizable from the menus, buttons, tools and pop-up menus because they are all controlled through Avenue programming. Through Avenue, the user is able to create ArcView add-on modules that are called extensions. The purpose of an extension is to preserve any Graphical User Interface (GUI) customizations and to provide quick access to customized functions or operations that the user creates.

The second goal of the present research study is achieved through Avenue programming to create custom functions and operations to simplify LA modelling. These customized functions are designed to allow the user to manipulate data from the Census Attribute Database in such a way that the data is used with a LA program. The results of the LA program are then subsequently mapped in ArcView using more custom developed tools. All of these tools are bundled into an ArcView extension that is called ArcView's Location-Allocation modelling extension, referred to as ARCLA hereafter. ARCLA is designed to improve the functionality of ArcView, so that it easily facilitates LA modelling.

Conceptual Integration Design

The integration procedure adopted by this study is very simple in its conceptual design. The Census Attribute Database works in conjunction with the GIS software through a database query that is performed using the RDBMS software that stores the Census Attribute Database. The Census Attribute Database query is then imported into ArcView, where the data are manipulated and exported in a format that the LA program accepts. Once the data are in the proper format, the LA program is run to find the optimal location for a range of facilities. After the LA program is finished, the solutions are then imported into ArcView where the results are manipulated and displayed using the graphical display abilities of the GIS software. Figure 5-2 graphically shows how the integration for GIS/LA system operates.

The intention of the ArcView extension is not provide one button that the user may use to run the whole process and map all of the LA solutions. Instead, the intention of the ArcView extension is to provide some specialized functionality that the user may use to set up the LA program and then map the LA solutions while still maintaining the maximum amount of flexibility for the user. In this type of arrangement, the user is able to choose the number of demand nodes and the number of potential facility locations and also select the solution(s) to map. This approach follows the same philosophy that the ArcView development team uses to develop ArcView. ArcView is designed to give the user a lot of flexibility. The user is presented with many functions in ArcView where each function performs a certain task and after the processing is performed by the function, it terminates to give the user the ability to choose what direction they would like to proceed in next.

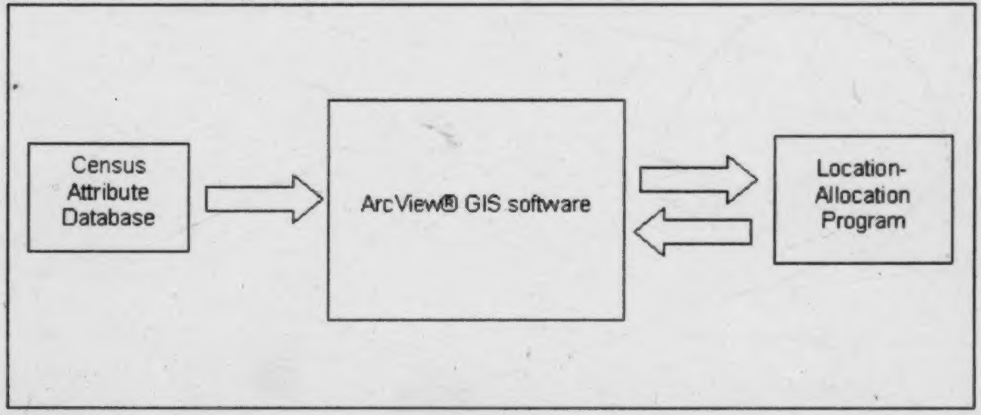


Figure 5-2 Conceptual integration design between the Census Attribute Database, ArcView and the location-allocation program.

The study's framework is based around using an existing LA program. This approach is taken because it is viewed as being easier to make a few modifications to an existing LA program and design the GIS tools around the requirements of the LA model rather than design a new LA program from scratch. The Teitz and Bart (1968) heuristic is used as the solution technique. Dr. M. J. Hodgson of the Department of Earth and Atmospheric Sciences at the University of Alberta has graciously donated his LA program (1997) for use with the present study. Hodgson (1997) coded the Teitz and Bart (1968) heuristic into the FORTRAN programming language and compiled the program to operate on a Microsoft DOS® platform. The LA program is a DOS based executable, which is a self-contained "runnable" computer program that is executed and run from the DOS command line window. The program receives all of its run parameters from two user created text files and all of the location information is read from two additional text files. One text file stores data regarding the potential facility locations and the other text file stores information about the location of each demand node, where the location information is based on each node's latitude and longitude coordinate.

The LA program is set up to read all of the necessary run parameters from text files, thus allowing the program to be executed from ArcView without having to enter any command line parameters. Since the LA program is a DOS based executable, the type of integration that is used by the current research study is to directly call the FORTRAN executable from a preset directory path using Avenue requests. The advantage of this approach is that the end user is required to only to use the ArcView GIS software to manipulate and display the data before and after the data is sent to the LA program.

Location – Allocation Files

The LA program requires four input files. Two of the files store data about where each demand node and potential facility node is located and the other two files store the run parameters information. The program reads all of the relevant input parameters and location data from the four text files. The following Table lists the required file name for each of the four files and what data must be stored in each file.

File Name	Required Data	Purpose of File
[variable name]_demand_nodes.txt	• EAuid field (unique identifier), • EALat field (latitude), • EALong field (longitude), • A Census variable that is used as the demand weight in the location program.	Records the locations for all demand nodes and the weight associated with each node.
[variable name]_potential_sites.txt	• EAuid field (unique identifier), • EALat field (latitude), • EALong field (longitude), (longitude), records all of the locations for all the candidate facility nodes.	Records the locations for all the candidate facility nodes.
[variable name]_Input_Parameters.txt	Three parameters separated by blanks: • Number of random starts, • Lowest end of *p* (number of facilities) range • Highest end of *p* (number of facilities)	Instructs the program how to execute by specifying the required input parameters. The rational behind this file is to eliminate any user input upon execution of the location-allocation program.
Optimize_name.txt	• Contains the [variable name] in the first column of the file.	Tells the program which variable is being solved for.

Table 5-1 Location-allocation program input files specifics.

The LA program requires that the first three files listed in Table 5-1 start with the same variable name. The component labeled as the [variable name] is to be replaced by the user with some type of logical name for their data. The [variable name] component is followed by the predetermined name that is outlined in Table 5-1. For example, the user may wish to use the word "children" as their variable name, which means the files would be called children_demand_nodes.txt, children_potential_sites.txt, children_input_parameters.txt and optimize_name.txt would contain the word "children". The [variable name] component name must be the same for all four files.

The third text file stores the input parameters information on how many random starts that the user would like the program to use to calculate the solution files and also how many facilities to compute. The more random starts the user uses, the better the final solution may be.

However, there is a disadvantage with allowing the LA program to run for a large number of random starts as the final solutions may take longer to calculate depending on the number of potential facility locations used. Next, the LA program requires the user to instruct it on how many facilities to locate. For the number of facilities the user would like to calculate, they are required to type in a range of facility numbers. For example, a range typed in as 3 5 results in solutions for 3, 4, and 5 facilities being calculated by the location program and 5 5 results in solutions for 5 facilities. The text file requires the above information in the following format: number of random starts, lowest value for the number of facilities and the highest value for the number of facilities with a single space being placed between each input parameter. For example, the first line of a parameter file that would be used to calculate 1 to 20 facilities for 100 random starts would be: 100 1 20.

The location program creates several solution files which store the solutions to all problems it calculated. For example, if the program is to calculate the optimal facility locations for 1 to 10 facilities, all 10 solutions are stored in the same file. By storing all of the solutions in one file, the process of importing the files into ArcView is simplified because ArcView possesses some excellent query tools to extract the individual solutions. In addition, a procedure where the user is required to import 10 separate solution files would be problematic and time consuming.

File Name	Contains the following data:
[variable name]_Demand_Allocations.txt	• P field - contains the solution number. For example, 1 relates to the solutions for one facility. • Facility field - contains the EAuid number for the facility that each demand node is allocated to for that particular solution. • Demand field - contains the EAuid number of the demand node for that particular solution. Weight field - the demand weights are carried over from the demand_nodes.txt file
[variable name]_Facility_Locations.txt	• P field - contains the solution number. For example, 1 relates to the solutions for one facility. • Facility field - contains the EAuid number for the facility for that particular solution.

Table 5-2 Location-allocation program solution files specifics.

After the LA program has terminated its computations, the solution files must be imported into ArcView to display the final solutions. Because the original data sent to the location program came from ArcView, the locational information for each node is already available and all that is required is that the solution files must be joined to an existing theme that contains the correct information. Since the solutions for all *p* values are stored in the same file, the solution file must be queried to extract the individual *p* values from each of the solution files. After the appropriate queries are performed, desire lines must be drawn between each demand node and the facility it is allocated to.

ARCLA Basics

The extension developed for this research is an attempt to show the location community that real world data are very easy to use and that GIS provide excellent tools for visualizing, manipulating, and querying spatial data for use with LA models. The extension is composed of 33 customized functions created using Avenue to automate the query, display and manipulation processes. ARCLA seeks to automate many of the query, display and manipulation procedures so that the location researcher can quickly and efficiently manipulate Census data sets and display the solutions from a LA model.

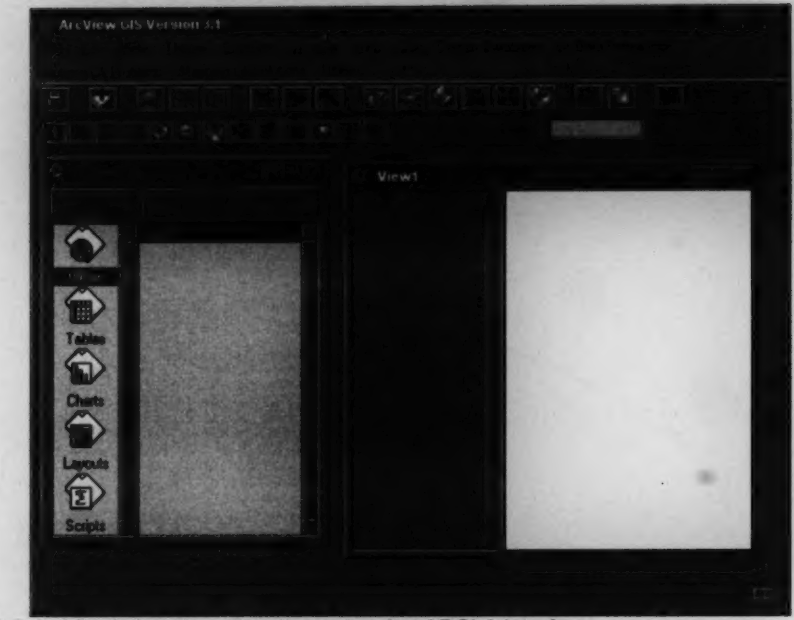


Figure 5-3 ArcView's location-allocation extension ARCLA interface.

The following sections explain how ArcView is modified to facilitate the LA automation procedure.

Census Attribute Database Integration

There are several methods available to access the data stored in the Census Attribute Database. However, only one method is recommended for extracting the attribute data from the Census Attribute Database. This method involves performing a query from within the RDBMS software and saving the results of the query as a file in database (dbf) format. The database (dbf) file is then exported from the RDBMS software for use by other software packages. There are several reasons why this method is recommended. First, the RDBMS can handle more complex queries than ArcView. Second, the RDBMS software allows the user to save and store all query statements for future usage or reference. Third, because the results of the query are stored in a separate file, these data are available at any time for analysis in ArcView. There are; however, a few disadvantages to this approach. The database and GIS are not truly integrated using this approach. In addition, the user is required to use the RDBMS software to extract the data and then import the resultant query into ArcView.

The database query should be performed for data that correspond to one Canadian City at a time. By only analyzing one city at a time, the likelihood of interchanging the files from several cities is reduced. This is particularly important with ArcView because ArcView is known to be poor for file maintenance. The Census database query requires the following information from the Census Attribute Database: the EAuid field values (EAuid), the corresponding latitude and longitude fields (EALat, EALong), the 1996 Total Population field (EAPop96), the city's name field (City_Name) and the desired Census demographic variable. The required Census demographic variable is used as the demand weight variable in the LA program and is derived from one of the demographic variables stored in the Census Attribute Database. The Census database query is saved in a dbf format and is referred to as the Census query table hereafter. Once the query has been completed and saved in the proper format, the user is now ready to initiate ArcView and begin the data manipulation procedure.

To start the data preparation process, the user must add the Census query table to the ArcView project. Once the Census query table is added to the project, the Census query table must be converted from a database file into an ArcView event theme. An event theme is a method of converting database tables into point themes based on some type of coordinate system such as X/Y or latitude/longitude. Since the Census query table contains the EALat (latitude) and EALong (longitude) fields, which define the centroid of each EA zone, the table can be converted to an event theme. The EALong field corresponds to the X field and the EALat field represents the Y field in ArcView. To automate the event theme conversion process, a program is included in the ARCLA extension to automatically turn the Census query table into a point theme. This function is found under the *Data Preparation* menu called *Convert Census Query to*

Point Theme. Once the Census query table is converted to a point theme, the user can begin the data manipulation process and create the files required by the LA program.

Creating the Demand Nodes File

The data manipulation process described in this section is entirely performed on the Census query table's point theme. Before the user can create the demand file, the user must identify the demand weight variable in the Census query table's point theme. The user can compute a new demand weight variable based on some aggregate variables such as *all the seniors*, or *all persons less than 5 years of age*. Once the demand weight field is identified, the user must prepare the data to be exported in ASCII text format. The LA program requires that the demand nodes file contain the following fields: EAuid, Latitude (EALat), Longitude (EALong) and a demand variable and only these fields.

The user must export the Census query table point theme in text format to a file using the predetermined naming convention. ARCLA provides the user with the ability to quickly create the demand file and properly name it. The function is called *Create Demand File for LA Program* and is located under the *LA Data Preparation* menu. The program is designed to automatically find the following fields: EAuid, EALat, EALong and it asks the user to select a field that is used as their demand weight. The program is designed to automatically name the export text file with the demand variable's name followed by the required "_demand_points.txt". For example, if the variable chosen is a field called 1067, then the program would call the export file 1067_demand_points.txt. The user may change the variable name of the export file if they wish, as long as they do not modify the "_demand_points.txt" component of the file name.

Creating the Potential Facilities Nodes File

As mentioned in an earlier section, all data manipulation for this section occurs on the Census query table's point theme. When creating the potential facilities text file there are several options available, as the user is able to use all EA centroids as potential facility locations or use some type of criteria to restrict the number of zones used. The user may want to simply study an individual neighborhood based on some type of site characteristics. For example, the user may want to study site characteristics of an area and they could restrict their data set to only those EAs with houses needing repair. Once the eligible facility locations are selected, they must be exported in text format for use with the LA program. The potential facilities text file requires the following fields: EAuid, EALat, EALong and only these fields. ARCLA contains a function that helps the user export the Census query table's point theme properly and saves the file, using the proper naming convention. The function is called *Create Facility File for LA Program* and is located under the *LA Data Preparation* menu.

Creating the Location-Allocation Parameter Files

ARCLA users can create the two input parameter text files from within ArcView with the function called *Create Parameters File for LA Program* located under the *Solving LA Problems* menu. This function displays a dialog box with 4 fields where the user is required to enter the relevant information to create the parameters text file as described above. The *Create Parameters File* function automatically creates the two necessary text files and populates them both with the appropriate information based on the data the user enters into the variable name text box.

Executing the Location - Allocation Program

Before the location program is run, the user must confirm that the four required text files are stored in the working directory. An ARCLA function allows the user to check that these four files exist and have followed the proper naming convention (variable_demand_points.txt, variable _potential_sites.txt, optimize_name.txt and variable_input_parameters.txt). The utility is called *Check LA Setup - search for required LA files* and is located under the *Solving LA Problems* menu. The program asks the user to input a variable name, this name should correspond to the variable name that was used on all four text files. If the four files exist and have the proper naming scheme, then a message box informs the user that the setup is acceptable and that the user is able to proceed by running the LA program. However, if one or more of the files are not named properly or are missing, the program instructs the user as to which file is missing or mislabeled.

To run the LA program, the user must ensure that the longav.exe is located in the C:\Temp directory. If the longav.exe is located in the C:\Temp directory, then the user is able to run the executable from within ArcView by using the function called *Solve Teitz-Bart Heuristic* under the *Solving LA Problems* menu. Once activated, the LA program does not prompt the user for any input as all of the parameters are handled by the information stored in the text files. When the LA program is finished, it produces five output files in the same working directory; however, only two of these files are important to the mapping component.

- **Mapping the Location-Allocation Solutions in ArcView**

After the LA program has completed its computations, the final solution files must be imported into ArcView so that the results can be queried and subsequently mapped using the ARCLA tools. ARCLA contains a menu called *Mapping LA Solutions* that is specifically designed to handle the tasks involved with importing the relevant solution files into ArcView and mapping the results. The two files of significance to this section are the files called [variable name]_demand_allocations.txt and [variable name]_facility_location.txt. There are two ARCLA programs of particular interest in order to import the solutions from the LA program. The functions are called *Import Facilities Solution File* and *Import Demand Solution File* respectively. Each program looks for the respective solution file either the [variable name]_facility_locations.txt or the [variable name]_demand_allocations.txt within the working directory. Since the solution files contain no location data such latitude/longitude, the files must be joined to the Census query table's point theme in order to associate the latitude and longitude coordinates for each node. A one-to-many join operation is required between the solution files and the Census query table's point theme because the solution file contains more than one solution for each EA zone. The join functionality is included in the import programs, so the join will automatically be performed for the user and the program will convert the joined table into an event theme so the user can see the data graphically.

The two solution files contain the entire solution sets for the requested facility ranges. For example, if the user calculated the optimal facilities for all facilities between 1 and 5, the solutions for all facilities are housed in the same solution file. Consequently, before the final solutions are displayed, the user must query the two solution files for all p values that correspond to only one particular solution. At this point, the user should have two new themes in their view, one that contains the solutions for all the facilities and the other that contains all solutions for the demand nodes. ARCLA contains two programs that aid in the query process: they are called *Query Demand Solution File* and *Query Facility Solution File*. The user must instruct the program as to which theme to query and for which value of p they would like to query.

Once the demand and facility themes have been queried for the respective values of p, the user may draw the desire lines between each facility node and demand node. This program is called *Create Desire Lines* under the *Mapping LA Solutions* menu and draws lines from each demand node to the facility that each demand node is allocated to. The program copies the demand weight field from the facility theme to the newly created desire line shapefile. As well, the program calculates the distance between each demand node and its corresponding facility node. The length field allows the user to quickly identify distances through a query in the theme's attribute table. Both the length and demand weight fields are useful for display and analysis purposes. Either of these fields can be displayed in ArcView, which allows the location researcher to increase his/her understanding of the LA solutions. When the program is finished, the user sees a new line theme, which connects each demand node to the facility it is allocated to with a line. The allocations are based on the information stored in the [variable name]_demand_allocations.txt file because this file contains the EAuid number for each demand node and EAuid number for the facility that each demand node is allocated to for each p solution.

ARCLA Utilities

ARCLA provides a few tools to enhance the graphic display of the final solution maps. A program is included to change the legend for each facility node to a unique colour and symbol. Another feature allows the user to modify the legend of the demand nodes theme into a graduated point symbol map using quantile classes. Both of these functions are found under the *Mapping LA Solutions* menu. A program is also included to allow the user to create choropleth maps of their demand variable from the Census query table.

Before a choropleth map can be made in ArcView, the user must add the Census EA polygon database for the city of interest to their ArcView project. The Census EA polygon database will form the basis of the background map displaying a distribution of the demographic variable analyzed during the LA analysis. If the user does not have access to the Census EA polygon database for the city of interest, they can perform what is referred to as a spatial query, which allows them to extract their city of interest from the spatial database file that stores all of the EA zones in Canada or that city's CMA spatial database file. To perform a spatial query, the user must extract only those polygons that correspond to the city of interest from the larger Census spatial databases. To do this, the user must join the Census query table (source table) to the attribute table for the Census spatial database (destination table), which is either the file with all EAs zones for the entire country or that particular city's CMA region polygon database. The common linking field for both tables is the EAuid field. Once the join procedure is complete, the user can query for all values that meet a certain criteria, such as all the records that correspond to the city name in the City_Name field. The result will be a highlighted selection of polygons that contains only the Enumeration Areas for the city of interest. The highlighted selection should then be converted into a new shapefile and saved using an appropriate name. These steps can be eliminated if the user has the selected city as an ArcView theme already. The key to the whole spatial query is the EA_Lookup table, the EA_ lookup table stores the information about which EAuid number corresponds to which city, thus make a spatial query of this nature possible.

To create a choropleth map, the user must use the city polygon database file created above and ensure that the Census query table is joined to the polygon theme's attribute table. A choropleth map is a thematic map portraying the distribution of a phenomenon. Choropleth maps are useful for visual interpretation purposes such as showing the overall distribution of a demand variable across an entire city. By creating a choropleth map, the user is normalizing or adjusting the values of one field (census variable) in order to show how these data relate to the values of another field thus making the data easier to read and understand (ArcView 3.1 On-line Help, 1998). It is recommended that the variable used as the demand weight be normalized by the total population (EAPop96). To aid the choropleth map making process a program is included in the ARCLA extension called *Create a Choropleth Map for Polygon Theme*. This function is located under the *Mapping LA Solutions* menu. The program automatically sets the legend to a graduated colour type with a quantile distribution using the number of classes established by the user and the classification variable is normalized by the second variable selected. The quantile classification method is used because each EA zone is composed of approximately the same number of dwelling units and a quantile classification is well suited to data that does not have disproportionate numbers of features with similar values (ArcView 3.1 On-line Help, 1998).

Maps are a two-dimensional representation of a roughly spherical surface; earth. In order to transform a three-dimensional space onto a two-dimensional map, a projection is required. A projection is a "...technique for transforming the three-dimensional sphere of the earth into the two dimensions of a map. There are four aspects of the map to be considered: area, distance, shape and it is impossible to recreate them all in the same map." (Mayhew, 1997, p. 343). In essence, projections try to preserve shape and area, while trying to minimize the effects of distortion. Therefore, this means that in order to calculate proper distance measurements in ArcView, the calculations must be performed on project data sets.

ArcView, by default, does not assign spatial data sets any projection parameters. The Statistics Canada data, on the other hand, are projected using a Lambert Conic Conformal projection. Since the Statistics Canada data are in the Lambert Conic Conformal projection, ArcView does not know how to calculate the proper distance measurements between each latitude and longitude pair in an unprojected View. Thus, this means that these data sets must be projected in order to obtain proper distance measurements.

Other ARCLA functions, such *as Add Lat., Long Coordinates to Point Theme*, require non-projected data sets, consequently the researcher decided not to impose any projection restrictions on the user in the ARCLA interface. Instead, the user is required to manually modify the projection parameters for the different ARCLA functions. Message boxes have been included in the ARCLA extension notifying the user if they need to modify any of the projection parameters before executing the function.

Up to this point, the data that are displayed in the GIS have not been properly projected, so the shape of the city is distorted. A utility provides ARCLA with the ability to project the Census spatial database files. The utility is called *Set the View's Projection* and is located under the *Utilities* menu. The program changes the projection to Lambert Conformal Conic with a spheroid of GRS 80 and uses the default Central Meridian supplied by Statistics Canada. The program also sets the map and distance units to meters, which corresponds to the Census spatial databases. Once the legends are modified and the data are properly projected, the user can create a layout of the view for presentation purposes.

The ARCLA extension includes other utilities to aid in the data manipulation and display process; however, these tools will not be examined in this chapter but are discussed further in Appendix 4.

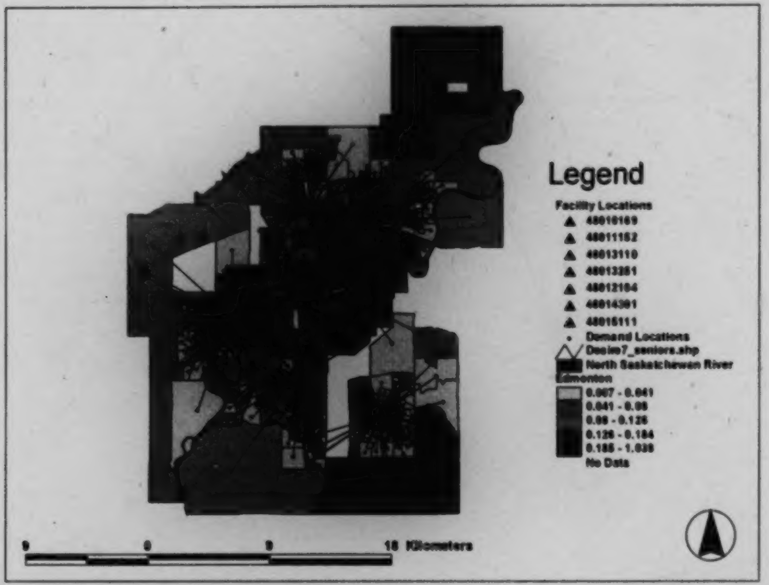

Legend
Facility Locations
48010169
48011152
48013110
48013281
48012104
48014301
48015111
Demand Locations
Dasim7_seniors.shp
North Saskatchewan River
Edmonton
0.007 - 0.041
0.041 - 0.08
0.08 - 0.126
0.126 - 0.184
0.185 - 1.038
No Data
9 0 9 18 Kilometers

Figure 5-4 Demonstration of the final output from the ARCLA extension. Image is displaying the location of five facilities for the senior population.

Enhancing ARCLA

Since, the *p*-median model is a discrete space model, it is able to accept any type of distance metric. The location program that is used with the present research uses a distance matrix that is calculated between each origin and destination and is calculated "on the fly" in the Central Processing Unit's (CPU) memory. Because, the *p*-median model is a discrete space model and is not concerned with the type of distance metric used, the model will accept a rectilinear, network, perceived or even Cartesian distance metrics.

Presently, ARCLA is designed to provide location solutions calculated using the Teitz and Bart (1968) heuristic. Hillsman (1984) demonstrates that many models have a similar structure and consequently a few modifications to the way the objective function coefficients are derived; many other location problems can be defined as special cases of the same general mathematical form of the *p*-median model (Hillsman, 1984, p. 305). For this reason, ARCLA could be used to solve other location problems such as the Maximum Coverage model, Set

Coverage model and the *p*-center model. ARCLA would require a few minor modifications to the location program executable where by the distance matrix would be manipulated to fulfill the requirements of the new model.

In essence, a Decision Support System is a set of tools that aid decision-makers by making their decisions more effective and efficient. With some enhancements, ARCLA could be modified so that it is a true Decision Support System (DSS). ARCLA could be modified and enhanced to be a fully functioning DSS by modifying how the user interacts with the GUI. One of the key modifications to ARCLA would be to minimize the data manipulation performed by the end user, thus making the extension easier to use. In order to reduce the number of steps required, a one-button interface would be the most effective solution. A one button system would control everything from exporting the necessary files, to running the location program, to importing the solution files back into ArcView and finally mapping the appropriate solutions. The one button interface would still require user input on many critical aspects. For example, the user would need to enter in the number of facilities that the location program requires to compute the solutions. As a DSS, the extension could offer the user the ability to change the setup parameters by adding or removing demand and/or facility nodes quickly by clicking on the selected node(s). This would allow the user to redefine the problem quickly and then calculate a new set of solutions. Another way ARCLA could be modified would be to include several different location models and user would need to select the model they wish to use to calculate the optimal solution. On the technical side, a DSS of this type would require tighter integration between the GIS/DSS and the RDBMS. All of the querying between the GIS and RDBMS would be handled "on the fly" and "behind the scenes". This would allow for faster queries and the user would not need to use the RDBMS for querying purposes. However, it is important that the system remains flexible even as a DSS.

Summary

This chapter demonstrates how ArcView can be modified to suit the requirements of a LA model by simplifying the processes required for data preparation, manipulation and data visualization. The data that are stored in the Census Attribute Database are a fundamental component to the whole process because these data allow location researcher to analyze real cities. To facilitate the use of the Census data, the LA program is designed to handle latitude and longitude coordinates, which allow the data to be exported from ArcView for use with the program. The LA program is also designed to read files that are exported from ArcView. After the LA program completes its computations, the results are imported into ArcView and joined to an existing point theme and then subsequently mapped using ARCLA GIS functions. The key to all of the integration is that the programs are used for their strengths and minor modifications are performed enhance the integration and ensure that data can be data passed from one program to the other. This chapter also shows how easily other tools for display purposes can be created and utilized to enhance the location researcher analysis.

Chapter VI. Location-Allocation Analysis

The previous two chapters discussed the development of tools to aid and enhance LA modelling through the use of a desktop GIS and the Census Attribute Database. In this chapter, the tools discussed in the previous chapters are used to perform a LA study for several Canadian cities to determine the effects of data surrogation in LA modelling.

This component of the current study is an extension of research conducted by Hodgson and Bayne (1990) where they analyzed one city, Edmonton, in terms of the accessibility implications of a multiservice center concept. They defined a multiservice center as a building that would service multiple uses such as a daycare center and senior's drop-in center. They looked at the City of Edmonton in terms of how well a multiservice center would serve two primary populations: the seniors and preschool children demographic groups. In order to analyze the accessibility effects, they first computed the optimal facility locations for each of the groups using the *p*-median model. Next, they analyzed the effects of using surrogate data to represent another demographic group. They used the facility locations for the total population (surrogate data) to serve the children's and senior's population. A measure was calculated to determine how much worse off (disadvantaged) the children or seniors population would be if they were served by a facility for the total population (surrogate) instead of the facilities specifically designed for them. This measure is a percentage value which indicated how much further a person (a senior or child) was required to travel using the system designed for others (surrogate data). Although it was not explicitly stated in their research, the research was, in essence, referring to data surrogation which is when one population is being served in a system based on the distribution of another group or surrogate population.

The present study is able to extend this research to several cities. With the tools developed for this study, a location problem for fifteen Canadian cities is undertaken to analyze the effects of data surrogation in LA modelling. The analysis for the present study is conducted using the framework laid out by Hodgson and Bayne (1990). First, the optimal facility locations are calculated for three demographic groups (preschool children, seniors and the total population). The demographic variables analyzed are children less than six years of age (preschool), seniors (individuals who are 65 years of age and older) and the total 1996 population for each EA zone. The reason the children and senior groups are chosen relates to their spatial distributions, these two groups have very different spatial distribution and as a result, these data are good to demonstrate the effects of surrogation. Secondly, the researcher calculates how disadvantaged the preschool children and elderly populations would be if their facilities were located using the facility solutions designed for the total population. After the optimal facilities locations for each of three demographic groups are calculated, a surrogation error index is determined to analyze the overall effects of using the solutions for the total population surrogate instead of the target group.

The analysis performed in this chapter, is a response to the aggregation problems encountered by location practitioners in the past. Spatial aggregation problems require the researcher to use the least aggregate data available; therefore, this means that the spatial resolution of the data are at the individual person level. An example of this would be data that represent individual households with a phone (this would allow a location researcher to avoid the effects of spatial aggregation). However, data of this nature are very difficult to obtain and are very difficult to work with since the size of the location-allocation problem would be very large. Another problem results from using the least aggregate data possible as the Census of Population data are not available at this spatial resolution. Therefore, location researchers often use a technique to reduce the size of the problems by reducing many demand locations to fewer demand locations. This is known as aggregation. This technique leads to aggregation error and is discussed in detail by Hodgson, Shmulevitz and Körkel (1997). This creates a dilemma for the location researcher because they need to decide if they should reduce spatial aggregation error by using data that are available at the individual person level. A problem occurs with this, if spatial aggregation is reduced, then attribute aggregation (or surrogation) is increased. Surrogation error arises when the spatial distribution of the proper population differs from that of the surrogate population.

There are several reasons for using surrogated data sets. First, it is difficult to find data that are highly disaggregate. Second, location problems should be spatially disaggregated, since these data are difficult to obtain, surrogation data will have to be used because data do not exist at the individual level. Consequently, the analysis that is performed in this chapter is an attempt to see if using surrogated data sets causes any detrimental effects to the solutions of a location problem.

The Location-Allocation Model

The *p*-median model is difficult to solve optimally, as a result, the Teitz and Bart (1968) heuristic is used. In order to find a good set of facility locations, the Teitz and Bart (1968) heuristic adheres to an improvement or 'hill-climbing' rule. The heuristic starts with an initial set of facilities (typically randomly chosen) and then the heuristic searches for ways to improve this configuration by reducing the objective function, through a series of transitions or interchanges. "A transition is defined as a swap between a facility location and a non-facility candidate-location, through a demotion of potential facility to be an ordinary candidate and the simultaneous promotion of non-facility candidate-location to be a facility." (Horn, 1996, p. 1700). The heuristic only performs the swap procedure if it will produce a reduction in the total objective function. The heuristic terminates if it can no longer reduce the total objective function for the problem. The resulting solution reached is called "...a local optimum, because it is not necessarily the least-cost, or globally optimal, solution for the problem." (Horn, 1996, p. 1700).

Rosing (1997) conducted a study to examine the relative effectiveness of the Teitz and Bart (1968) heuristic method with increasing numbers of demand nodes and facilities locations. His results indicate that there is a definite reduction in the effectiveness of the heuristic with increasing demand nodes and facility values. This is not viewed as a major limitation on the present study as the analysis for the present study is being conducted using 100 random starts, thus expecting to avoid particularly bad solutions. By allowing more than one random start, the GIS/LA system developed for this research differs from the commercially available location applications. The reason for allowing more than one random start is to reduce the chances of the solution falling into a situation of a local minimum. This sets the GIS/LA system developed for this research apart from the presently available commercial applications. Moreover, the analysis is more for demonstration purposes of the tools developed rather than developing new solution methods or techniques.

Location-Allocation Problem Setup

The criterion used to select the fifteen Canadian Cities involved selecting only major metropolitan cities that are located in a Statistics Canada CMA region. This is because the attribute data for a chosen city are contained within the Census Attribute Database and the spatial component is available in the CMA spatial database files. The researcher tried to select cities from each region of the country: the West Coast, Prairies, Ontario, Quebec and the Maritimes. The following Table (6-1) shows the fifteen selected cities and their respective population breakdowns for the preschool children, elderly and the total population demographic groups including the percent of the total population that each demographic group consists of.

City	Total Population	Children Population	Percent of Total Pop.	Seniors Population	Percent of Total Pop.
Calgary	768,082	62,615	8.15%	68,335	8.90%
Edmonton	616,306	49,135	7.97%	67,690	10.98%
Halifax	113,910	6,640	5.83%	17,155	15.06%
Hamilton	322,352	25,000	7.76%	48,600	15.08%
North York	548,950	40,245	7.33%	85,160	15.51%
Ottawa	323,340	21,360	6.61%	49,420	15.28%
Quebec	167,264	9,315	5.57%	26,545	15.87%
Regina	180,400	14,820	8.22%	21,560	11.95%
Saskatoon	193,647	17,060	8.81%	22,345	11.54%
Scarborough	558,960	44,770	8.01%	66,795	11.95%
Surrey	304,477	28,650	9.41%	31,060	10.20%
Toronto	653,734	42,010	6.43%	78,380	11.99%
Vancouver	514008	27,795	5.41%	66,245	12.89%
Victoria	73,504	3,505	4.77%	15,645	21.28%
Winnipeg	618,477	48,325	7.81%	84,600	13.68%

Table 6-1 Population breakdowns for the fifteen selected Canadian cities. It should be noted that the population values are tabulated by Statistics Canada and may not be entirely accurate because of data rounding and suppression techniques.

After the cities are selected, the attribute data that correspond to the preschool children, seniors and total population are extracted from the Census Attribute Database for each city. The results of the Census Attribute Database queries are exported from the database and the resulting files are imported into ArcView. The ARCLA extension is used to manipulate the Census Attribute Database queries to create the necessary demand and facility files for use with the LA program. The demand and facility files are created for each demographic group; preschool children, seniors and the total population. In order for an EA zone to be used as a demand node, the zone must contain a preschool children's or senior's population, otherwise the EA zone is not included in the demand nodes file. Potential facility locations, on the other hand, are identified as all EA zones in a city. The LA program is set up to calculate the optimal facility locations for a range of facilities from one to twenty five facilities. The LA program is set up to use one hundred random starts, to reduce the likelihood of a solution producing a poor local minimum. The program is subsequently run for each demographic group for all of the fifteen cities. The following Table (6-2) shows the breakdown and size of the location problem for each city.

	Seniors		Children		Total Population	
City	Potential Facility Nodes	Demand Nodes	Potential Facility Nodes	Demand Nodes	Potential Facility Nodes	Demand Nodes
Calgary	942	859	942	794	942	916
Edmonton	826	757	826	682	826	792
Halifax	206	173	206	149	206	191
Hamilton	440	423	440	385	440	437
North York	795	698	795	630	795	719
Ottawa	533	471	533	404	533	512
Quebec	313	281	313	224	313	303
Regina	237	220	237	200	237	229
Saskatoon	266	245	266	213	266	256
Scarborough	704	677	704	627	704	694
Surrey	350	342	350	319	350	348
Toronto	1188	1095	1188	962	1188	1160
Vancouver	860	774	860	594	860	833
Victoria	169	149	169	106	169	154
Winnipeg	862	806	862	705	862	847

Table 6-2 Location-allocation program setup statistics.

GIS Analysis

Choropleth maps reveal the roots of the surrogation problem as they depict the population's distribution per capita. Choropleth maps for each city are created for the preschool children and elderly demographic groups (choropleth maps for the other fourteen cities are located in Appendix 5). The choropleth maps show that the two demographic groups have very different distributions in all of the fifteen selected cities. The elderly population (Figure 6-1) are typically concentrated closer to the city center (core) and the preschool children (Figure 6-2), on the other hand, are located in the suburban regions of the city further away from the city center. The senior's population also displays a spatial pattern with lower concentrations in the suburban area and higher numbers in the central core (Figure 6-1). Children, on the other hand, exhibit high numbers in the suburban areas and lower numbers in the central core (Figure 6-2). Overall, the total population does not show a strong spatial pattern or distribution in Edmonton (Figure 6-3). The choropleth maps for the seniors and children groups reveal that that the populations for these two groups vary more than the total population.

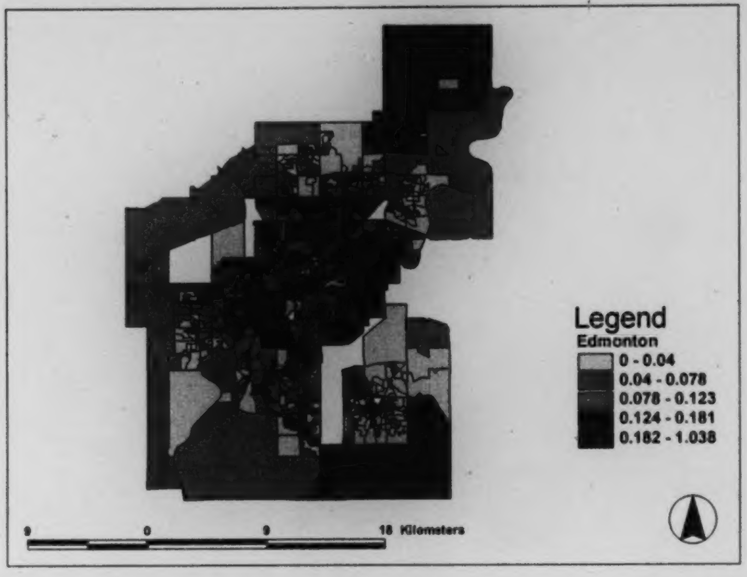

Legend
Edmonton
0 - 0.04
0.04 - 0.078
0.078 - 0.123
0.124 - 0.181
0.182 - 1.038
9
0
9
18 Kilometers

Figure 6-1 Senior's population per capita in Edmonton.

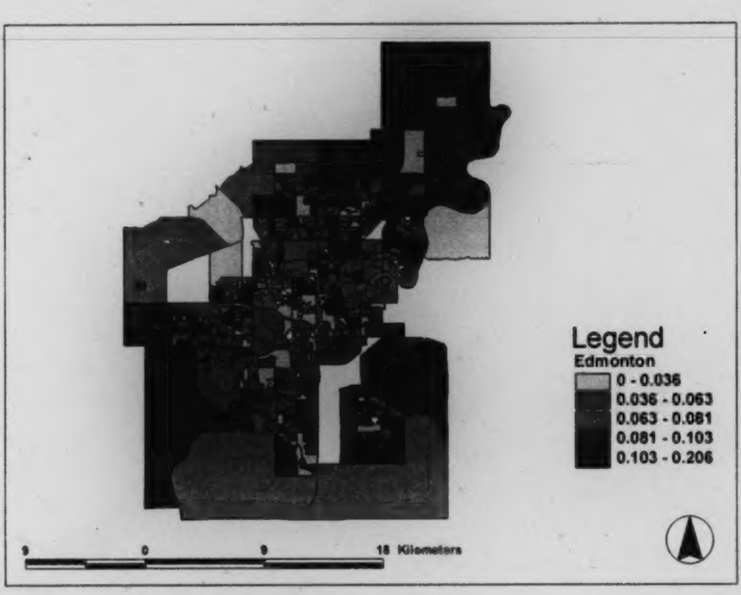

Legend
Edmonton
0 - 0.036
0.036 - 0.063
0.063 - 0.081
0.081 - 0.103
0.103 - 0.206
9
0
9
18 Kilometers

Figure 6-2 Children's population per capita in Edmonton.

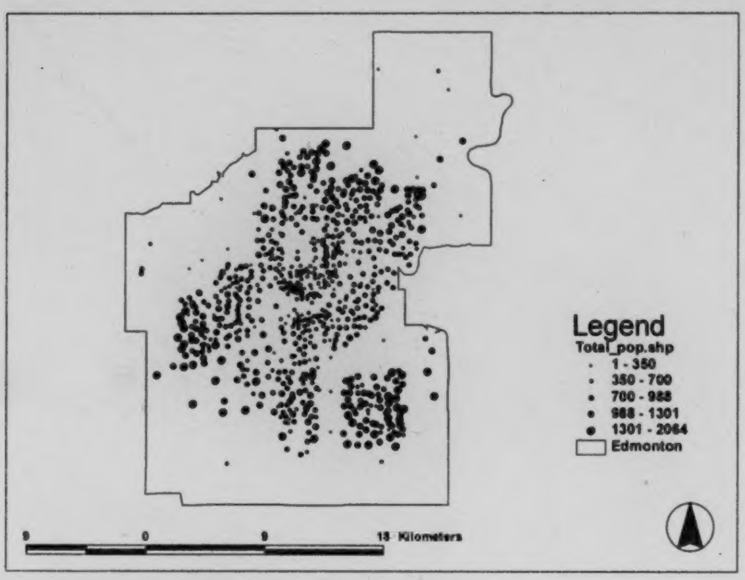

Legend
Total_pop.shp
1 - 350
350 - 700
700 - 988
988 - 1301
1301 - 2064
Edmonton
9
0
9
18 Kilometers

Figure 6-3 Total population distribution in Edmonton.

The distribution patterns of these two groups are also evident in the other fourteen Canadian cities analyzed. All of the other cities show strong children populations in the suburban areas and a strong senior's distribution near the central core.

Once the solutions were available, the researcher was able to import the results into ArcView for display and analytical purposes. The following Figures (6-4, 6-5 and 6-6) show the allocations for seven optimally located facilities for the preschool children and elderly demographic groups in Edmonton.

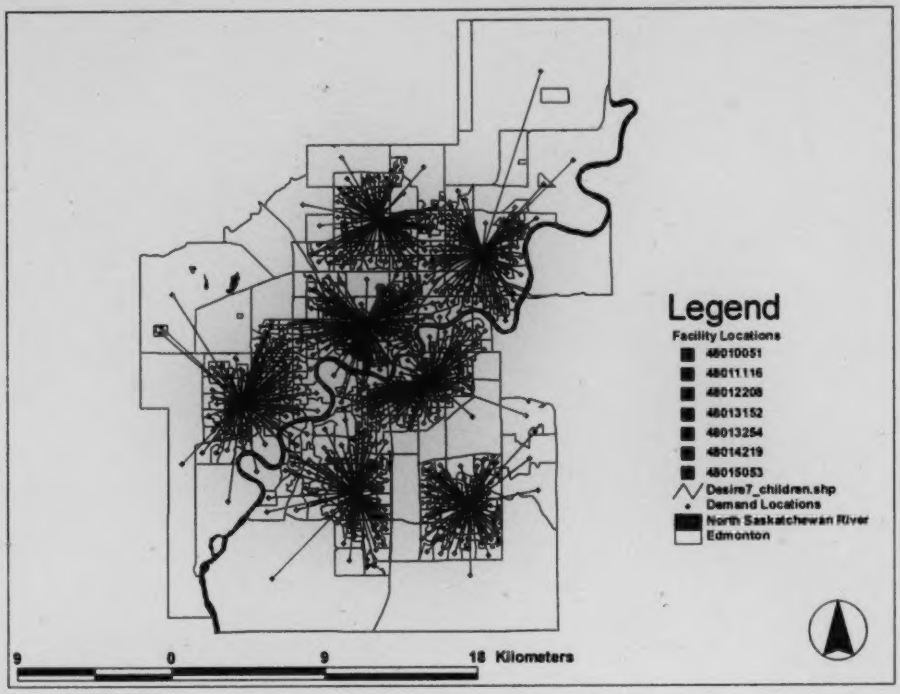

Legend
Facility Locations
48010051
48011116
48012208
48013152
48013254
48014219
48015053
Desire7_children.shp
Demand Locations
North Saskatchewan River
Edmonton
9
0
9
18 Kilometers

Figure 6-4 Final facility allocations for seven preschool children facilities (less than six years of age) in Edmonton.

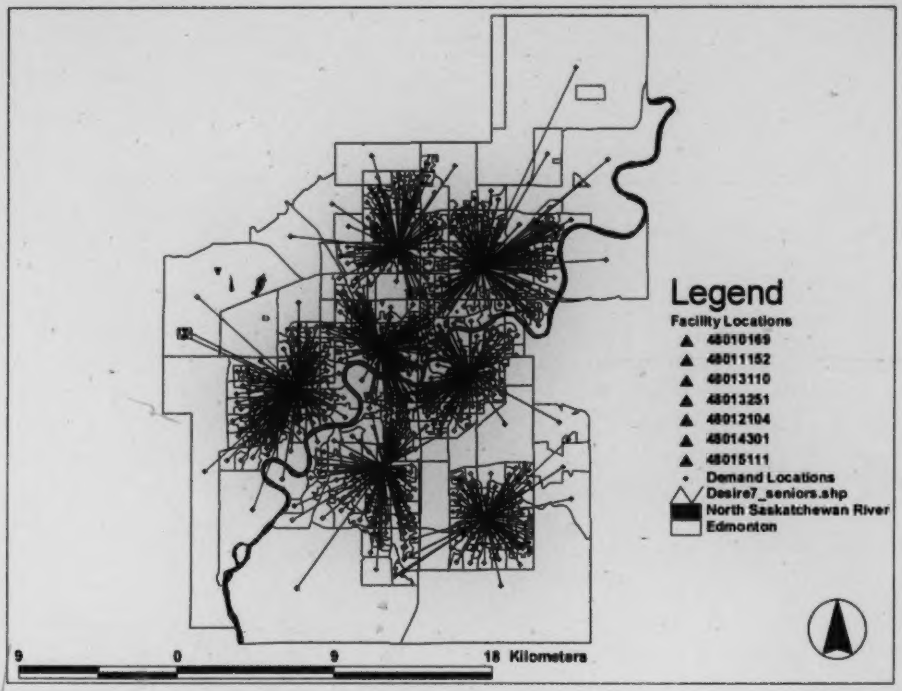

Legend
Facility Locations
▲ 48010169
▲ 48011152
▲ 48013110
▲ 48013251
▲ 48012104
▲ 48014301
▲ 48015111
• Demand Locations
Desire7_seniors.shp
North Saskatchewan River
Edmonton
9
0
9
18 Kilometers

Figure 6-5 Final facility allocations for seven senior facilities (65 years of age and older) in Edmonton.

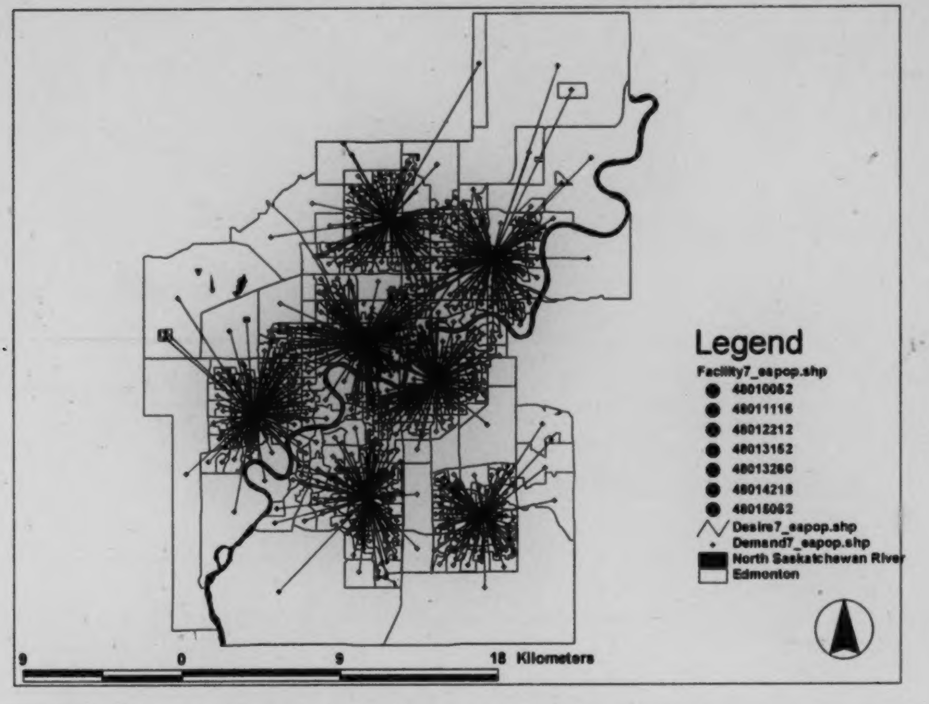

Legend
Facility7_eapop.shp
● 48010062
● 48011116
● 48012212
● 48013152
● 48013260
● 48014218
● 48015062
Desire7_eapop.shp
Demand7_eapop.shp
North Saskatchewan River
Edmonton
9
0
9
18 Kilometers

Figure 6-6 Final facility allocations for seven total population facilities in Edmonton.

A visual comparison of the previous three Figures that portray the facility locations for both of the demographic groups and the total population reveal interesting results. It becomes apparent that there are variations between the three facility location structures. The facility system for the total population shares only one common facility location with the children's system and this indicates that it would not be beneficial to either group if they are served by the facility system for the total population. In essence, Figures 6-4 and 6-5 show that each group would not be well served by a different facility system.

In each of these systems, the average weighted distance decreases with an increasing number of facilities and can be displayed in a cost-effectiveness diagram. Figure 6-7 exhibits this cost-effectiveness characteristic. The cost effectiveness graph for Edmonton and the other fourteen cities show that the senior's population is easier to serve with a smaller number of facilities due to their spatial distributions. The senior's population is typically centrally located and is easier to serve with fewer facilities than the children's population. Initially for one facility, the average weighted distance for the senior's system is smaller than that of the children's population; however, as the number of facilities increase the two systems generally have similar average weighted distance values that they must travel to their respective facilities. This is because as the number of facilities increase the effects of spatial distribution are reduced. The children's population has a larger average weighted distance for fewer facilities because of their spatial distribution. The children are not as centrally located as the seniors, consequently their facilities are further away because the system is less concentrated (the children tend to be located in suburban areas). Figure (6-7) depicts the cost-effectiveness graph for both the senior and children's population in Edmonton. The cost-effectiveness graphs for the other fourteen cities are found in Appendix 6.

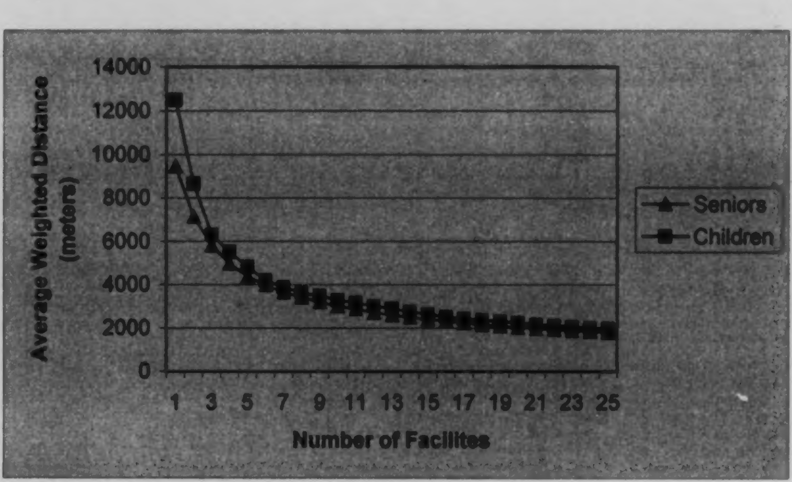


Figure 6-7 Cost-Effectiveness graph for Edmonton.

Surrogation Error

In this section, the researcher analyzes how disadvantaged one of the two demographic groups would be if the facility system used is derived from the surrogate population, instead of the system specifically designed for a particular group. The comparison is performed by determining the surrogation error, which evaluates the suboptimal solution in terms of how suboptimal the solutions are when using a surrogate population in place of the specific population's solutions. It is imperative to remember that the overall goal of a LA model is to minimize the aggregate distance between the demand nodes and the nearest facility to them. For this study, a surrogation error method is devised to calculate a percentage error measure that arises from the use of surrogate data sources. In essence, this means that the solutions for the surrogate population are used in place of the specific group's solutions. The surrogation problem is defined by the following terms:

Λ_S	is the optimal set of facilities found using the surrogate demands (the total population solution in this study)
Z_{SP}	is the objective function evaluated using Λ_S and proper groups (correct) demands.
Λ_P	is the optimal set of facilities found using the proper groups demands (where the population is either the children's or senior's solution for this study)
Z_{PP}	is the objective function evaluated using Λ_p and the proper group's demands.

In percentage terms, location error is calculated as:

$$\textit{Surrogation Error} = \left[\frac{Z_{SP} - Z_{PP}}{Z_{PP}} \right] * 100$$

The surrogation error value represents the degree to which a child or senior would be required to travel further to use facilities that are designed for the total population than to facilities designed specifically for them.

The present study uses the optimal facilities for the total population to serve the two preschool children and seniors demographic groups. Figure 6-8 reveals what happens when the total population are used instead of using the data for the preschool children or seniors in Edmonton. The graph shows how much further an individual in either the elderly or preschool children's demographic group would have to travel if the facility was allocated for the total population.

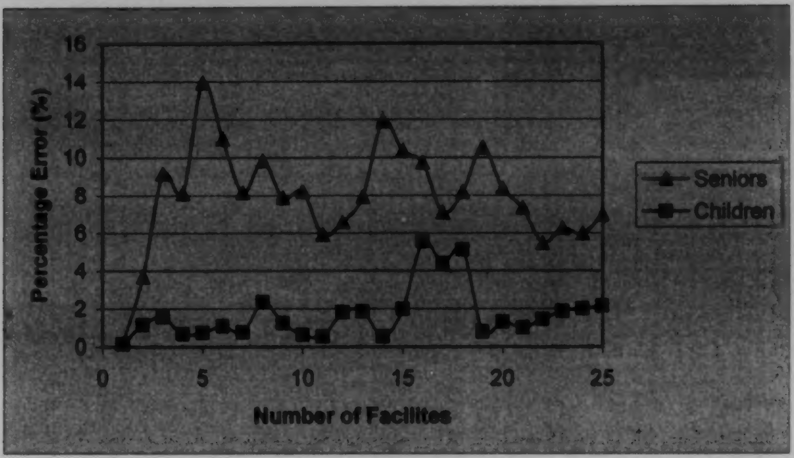


Figure 6-8 Location Error - children and seniors (proper population) demographic groups compared to the total population (surrogate data) for Edmonton.

In this arrangement, both populations are reasonably well served for one facility, but the surrogation effects increase with an increase in the number of facilities. The graph in Figure 6-8 shows that for five facilities, the children are 0.74% more disadvantaged and the elderly are 13.9% more disadvantaged if they had to travel to a facility in this configuration. Figure 6-8 exposes the effects of data surrogation error and demonstrates that data surrogation does have an overall effect on the seniors and children's populations.

The next two Figures (6-9 and 6-10) display the minimum, maximum and mean surrogation error values for the seniors and children's populations for all fifteen Canadian cities. The minimum surrogation error reveals that surrogation error exists in all solutions. The maximum surrogation error values show that the error follows a linear upward trend as the number of facilities increase. Typically, maximum surrogation error values are highest for twenty-five facilities.

The surrogation error is greater for the senior's population than the children's population. As shown in the cost-effectiveness graphs the senior's population had greater benefit with lower number of facilities. In the system when the seniors use the total populations facilities, the average weighted distance values are increased meaning the seniors must travel further and this yields a higher surrogation error. This may be explained by the distribution of the senior's population as they tend to be more centrally located and the facility system for the total population does account for the senior's central distribution. The children, on the other hand, have similar travel distance values as the total population; consequently, the surrogation error values are lower than the seniors. This indicates that surrogate populations yield poorer results when they are used to represent a certain target population group like the preschool children and seniors.

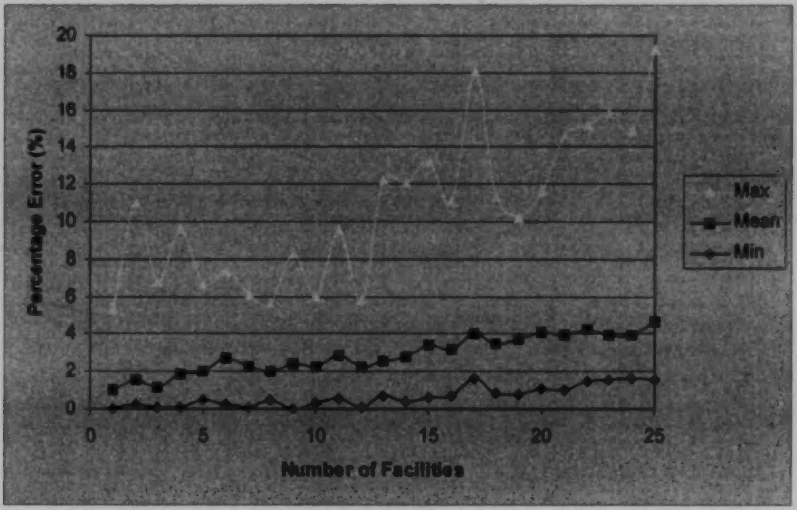


Figure 6-9 Surrogation Error - seniors (proper population) demographic group compared to the total population (surrogate data) for all fifteen cities.

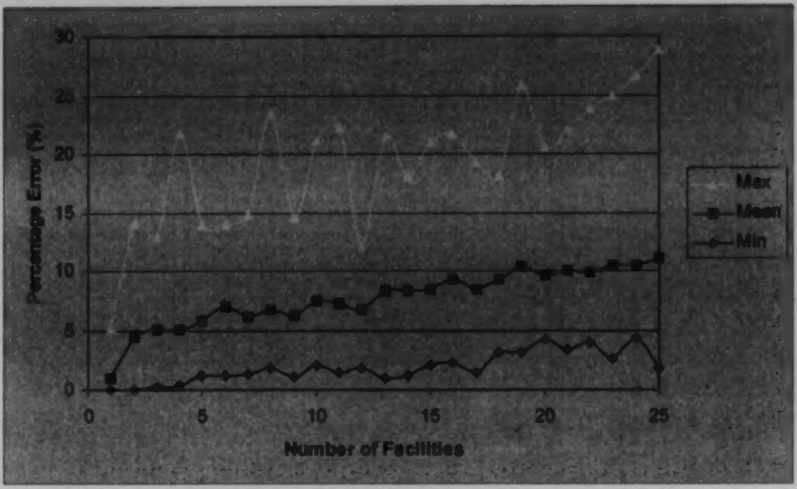


Figure 6-10 Surrogation Error - children (proper population) demographic group compared to the total population (surrogate data) for all fifteen cities.

Of the fifteen cities examined, surrogation error for Toronto is the least severe for both the preschool children and the senior demographic groups as the error values are below 6%. The surrogation error problem is most severe for the City of Victoria where the error is typically over 15% and as high as 19.7%. It can be hypothesized that the consistently poor results for the city of Victoria are a result of the city's structure as there is a wide disparity between the percentage of the total population that makes up the preschool and seniors groups. Victoria is also surrounded be several small suburban communities and it is evident that these communities have and effect on the surrogation error. By far, the poorest surrogation error for one facility was achieved for the senior's population in Saskatoon. The location error for twenty-five facilities in Saskatoon was 28.7%, thus indicating that the senior population would not be well served if the facilities were located using the total population's facility system.

Summary

This chapter demonstrates that it is very feasible to use GIS tools and the Census Attribute Database for LA analysis. These tools allow the location researcher the ability to quickly access data to perform analysis on many cities. The research undertaken in this chapter computes the optimal facility locations for two demographic groups (preschool children and seniors) and then substitutes their facility solutions with the facility solution for a surrogation population to determine what error is created by using these surrogate data. In the last section of this chapter, the solutions for the two demographic groups are compared to a surrogate population (total population) to determine if any surrogation error exists in the facility locations for the two demographic groups analyzed. The results indicate that surrogation error exists in all cases, no matter how small it is. Overall, the surrogation error varies from city to city, but it is present for each city. One trend that is observed from the surrogation results for the fifteen cities are that the surrogation error seems more pronounced for the senior population than the children population. Consequently, the researcher can conclude from this analysis that surrogation error does exist and that it arises from using data from different sources and does ultimately affect the solutions in LA modelling.

Chapter VII. Summary and Discussion

In concluding this research, it is appropriate to reflect on the two major objectives of the study that are outlined in the introduction. The first objective is to demonstrate the ease with which a GIS and a Census Attribute Database can be integrated to provide quick and efficient access to real world data. The purpose of the database is to serve as a data bank or warehouse of socio-economic variables that are useful for location-allocation problems and problems involving socio-economic data analysis.

The second objective is to demonstrate how a desktop Geographic Information System (GIS) and a location-allocation model can be integrated to take advantage of the strengths of both applications.

First Objective

Initially, a problem was encountered during the research phase on how the 1996 Canadian Census of Population data could be used in order to facilitate the extraction of socio-economic variables for an individual Canadian city. When the 1996 Census of Population data are in their original format from Statistics Canada, the data are very difficult to use in terms of how to extract data pertaining to a particular Canadian city. Consequently, an EA lookup table was developed to resolve this problem. The lookup table contains all of the EAuid values for all EA zones that are contained in any of the 1996 Census CMA spatial database files. During the creation of the EA lookup table, each EAuid number was entered in the table so that each zone is coded to correspond to the city that the zone belongs to. Coding each EAuid to its city name allows for a query that can extract data for any major metropolitan city in Canada. After the EA lookup table was coded, the nine Nation Tables and the EA lookup table were imported into a commercial RBDMS software package where the Census Attribute database was created by joining all nine nation tables to the EA lookup table.

In order to facilitate the integration between the Census Attribute Database and the desktop GIS software, the EALat and EALong fields were added to the EA lookup table. The EALat and EALong fields correspond to the latitude and longitude coordinates for each EA centroid. These two fields are important because a query that is performed in the Census Attribute Database can be imported directly into the GIS software without having to be concerned about any data conversion or manipulation issues.

With respect to the first objective, several conclusions can be made. First, the Census Attribute Database including the EA lookup table is an indispensable component of this research as it provides a method to quickly and efficiently access socio-economic data for any major metropolitan city in Canada. The Census Attribute Database also provides the ability to extract data for any metropolitan city from the 1996 Census spatial database files. Secondly, once the database was developed, it can be continually used for other research ventures and decision making processes. The Census Attribute Database is important because it offers over 1500 Census demographic variables to a location researcher for all types of location-allocation research problems. One example of this is demonstrated and discussed in chapter 6.

Second Objective

The second objective was achieved by utilizing what is termed as loose coupling integration approach. This means that each tool is used for its strengths and the integration between each application occurs through the sharing of data between all applications. In order for the integration to work properly, each application was modified in such a way that each can accept and return the data back to the other application. The desktop GIS is used for its data display and data manipulation abilities. The location-allocation program is used to compute the optimal locations of a set number of facilities. The RDBMS is used to store the Census Attribute data and for its powerful query abilities.

To facilitate the integration, the desktop GIS was modified through the use of its object oriented programming language, Avenue, which allows for the development of GIS/LA tools.

The modifications to the GIS were performed around the requirements of a location-allocation model. The intention of the integration component of this research is to demonstrate how easily the integration can be done while minimizing the amount of complexity. The integration is simple in its design, as data from the database are passed to the GIS software where the data are in turn manipulated and exported in a format acceptable to the location-allocation program. After the location-allocation program has finished its computations the data are imported back into the GIS application where the individual solutions can be mapped. The location-allocation program requires several input data files. One file is used to indicate, through latitude and longitude coordinates, the locations of each demand node, another file is used to store the potential facility locations and two setup files are used to eliminate the need for run time parameter commands. After the location-allocation program is completed, the solutions are stored in two text files, which must be imported back into the GIS software to display the solutions.

This component of the research shows how easily a location-allocation model can be integrated with a GIS and RDBMS for location problems with real world data and for analyzing the socio-economic conditions present in several Canadian CMAs. Each software application is being using for its strengths and thus makes the whole system very easy to use and set up. The ability to quickly access and process the data before they are sent to the location-allocation model is one of the fundamental strengths of this study. Consequently, data are no longer cumbersome and time consuming to work with for location-allocation modelling and socio-economic data analysis. The only substantial modification to the location-allocation model involves setting up the program to read and calculate distances based on latitude and longitude coordinates.

Several conclusions can be reached from this objective of the research. First, the integration of a GIS and location-allocation model is very feasible and has very beneficial outcomes. The desktop GIS offers spatial query analysis tools, it is able to effectively handle real world data sets and through some modifications, the GIS can offer the user cartographic tools such as choropleth mapping, where the data are visually plotted on a Census boundary map. GIS offers excellent and easy to use tools, plus the costs of a typical desktop GIS are more affordable now to the general population.

Data Analysis Section

This chapter is an important component of this study as it provides clear justification to the ease with which data for several Canadian cities can be analyzed in a location-allocation problem. The current research is able to do what few studies have done before to provide quick and easy access to data and then these data sets can be used to formulate a location-allocation problem for more than one city. The original location-allocation problem is derived from Hodgson, Bayne (1990) which looks at data surrogation for one Canadian city. The present study is concerned with the development of GIS tools, which allows research problems of this nature to be conducted on a larger scale, mainly for many cities. The results of the analysis section for the present study which looks at data surrogation for fifteen Canadian cities reveals that some surrogation error, no matter how small, exists for all of the cities analyzed. The error does vary considerably from city to city for the fifteen Canadian cities selected. The surrogation error seems more pronounced for the senior population than the children population.

The present research demonstrates that with the tools that are developed for this study, the location researcher now has the ability to formulate location-allocation problems for several Canadian cities instead of the commonly used one city. An example of such a study was conducted by Hodgson and Payne (1990) where they analyze surrogation effects for only one city, Edmonton. Previously, analysis of this nature would have been very difficult to perform. GIS analysis is only as good as the data that serves it, and for this study, once the Census Attribute Database was designed, it provided an excellent basis for location-allocation analysis. Although some data quality difficulties are identified in the study, such as data rounding and suppression, these data are still very valuable to location-allocation decision making process.

Applications of the GIS/LA System

The GIS/LA system that was developed for this research uses the Census of Population spatial and attribute data in such a manner that it facilitates quick and efficient access to any of the 1500 census variables for analysis or modelling problems. As a result, the overall GIS/LA system has the ability to be beneficial to a wide ranging audience, either in its present form or in the form of a one button Decision Support System. The applications of this GIS/LA system are numerous.

Urban planners will find the system useful for planning and forecasting purposes. They can use the census spatial data and attribute data sets to track temporal trends by analyzing the characteristics of municipal neighbourhoods. Demographers and social scientists will find the system useful for analysis because the system contains data for all of the Canadian CMAs. The database and GIS system are a wonderful combination that allows a Demographer to quickly analyze the characteristics of an area without a great deal of data manipulation. Business analysts will find the GIS/LA system to be an excellent tool for enhancing analysis for market surveys and for analyzing customer profiles. Finally, location-allocation practitioners will find the system extremely useful, as they will now have a mechanism to integrate real world data sets into their research applications.

Conclusion

The overall purpose of this research is to show how easily a location-allocation model can be integrated with a GIS whereby the GIS would be used as a decision support tool for a location-allocation model. GIS are excellent for organizing, maintaining and visualizing geographic data such as the 1996 Census of Population data sets. Desktop GISs are able to quickly and efficiently process data for use with a location-allocation model. GISs also have the potential to do much more. They can be used to do what the GIS practitioners call spatial queries, for example, selecting areas within a distance of 5 km of the facility and reporting back the number of people that live in those zones. The decreasing costs and increased user friendliness of

GIS software packages are also permitting more individuals to access these precious decision making tools for use with location-allocation problems. By using a GIS to do the labour intensive work, such as the data preparation with real world data sets, more effort can be directed toward developing and testing new location models. It is hoped that this research is just the beginning of advancements in the use of real world data and desktop GISs for location-allocation model analysis and problem solving.

Further Discussion

It would be interesting to examine how well the Teitz and Bart (1967) algorithm would perform if it is programmed in Avenue. It would provide a truly integrated environment where the model and the GIS are bundled in the same software package and the GIS could be used for more than just data manipulation and display purposes.

It would also be interesting to use other location-allocation models with the Census Attribute Database and the ARCLA GIS extension, to examine other location-allocation problems. Through a few modifications, an existing location-allocation program could accept the data that is generated from the GIS and the Census Attribute Database.

Statistics Canada needs to have better data retrieval techniques in order to extract data for a particular city. For example, Statistics Canada needs to develop an EA lookup table for use with the 2001 Census of Population and grant the general public access to these data. Such a data file would have made this research must easier to implement and would have dramatically reduced the development time associated with the development of the Census Attribute Database.

On a final note, further work should be undertaken to determine if surrogation error is more serious than aggregation error. The tools presented in this study would make analysis of this type very interesting and the research problem completely feasible.

VIII. Bibliography

Bennett, D. A. (1997). A framework for the integration of geographical information systems and modelbase management, International Journal of Geographical Information Systems, (11 (4) pp. 337-357). London, England: Taylor & Francis

Bradley, G. H. (1989). Mathematical Programming Modeling Project - Overview. In Sharda, Ramesh, Bruce L. Golden, Edward Wasil, Osman Balci, William Stewart (Eds) Impacts of Recent Computer advances on Operations Research. (pp. 447-462). New York, New York: North Holland.

Burgess, E.W. (ed.) (1923). The Urban Community. Chicago, Illinois: University of Chicago Press.

Burrough, P.A., & McDonnell, R. A. (1986). Principles of Geographical Information Systems for Land Resource Assessment. New York, New York: Oxford University Press.

Carver, S. J. (1991). Integrating multi-criteria evaluation with Geographical Information Systems, International Journal of Geographic Information Systems, (5 (3) pp. 321-339). London, England: Taylor & Francis.

Chang, N., & Lin, Y.T. (1997). Optimal-siting of transfer station locations in a metropolitan solid waste management system, Journal of Environmental Science Health. (32 (8) pp. 2379-2401). New York, New York: M. Dekker.

Chen, J., & Tang, Z. (1994). Spatial facility location analysis in GISs, The Canadian Conference on GIS - 1994 Proceedings International Conference Volume 1. (pp. 309-315). Ottawa, Ontario : Canadian Institute of Geomatics.

Chou, Y. (1996). Exploring Spatial Analysis in Geographic Information Systems. Santa Fe, U.S.A.: Onword Press.

Chuvieco, E. (1993). Integration of linear programming and GIS for land-use modeling, International Journal of Geographical Information Systems, (7 (1) pp. 71-83). London, England: Taylor & Francis.

Davies, W. K.D., & Murdie, R. A. (1991). Consistency and differential impact in urban social dimensionality: intra-urban variations in the 24 metropolitan areas of Canada, Urban Geography. (12 (1) pp. 55-79). Silver Spring, Maryland: V.H. Winston & Sons

Environmental Systems Research Institute (1998) Online Help ArcView 3.1

Goodchild, M., Haining, R., & Wise, S. (1992). Integrating GIS and spatial data analysis problems and possibilities, International Journal of Geographic Information Systems, (6 (5) pp. 407-23). London, England: Taylor & Francis.

Harris, C. D., & Ullman, E. L. (1945). The Nature of Cities, Annals of the American Academy of Political and Social Science (143 pp. 7 -17).

Haug, W. (1993). L'utilization du recensement de la population dans un systeme d'information geographique: l'example de la Suisse. Berne, Switzerland: Federal Statistical Office, Mimeo.

Hillsman, E.L. (1984) "The p-median structure as a unified linear model for location-allocation analysis," Environment and Planning A,. (16 pp. 305-318). London, England: Pion.

Hodgson, M. J., & Bayne, P. (1990). Accessibility implications of the multi-service centre concept for Edmonton. In Smith, P.J, E.L. Jackson (Eds.), A World of Real Places: Essays in Hour of William C. Wonders, (pp. 23-33). Edmonton, Alberta: Department of Geography, University of Alberta.

Hodgson, M. J., Shmulevitz, F., & Körkel, M. (1997). Aggregation error effects on the discrete-space *p*-median model: The case of Edmonton, Canada, The Canadian Geographer, (41 (4) pp. 415-28). Toronto, University of Toronto Press: Canadian Association of Geographers.

Hodgson, M. J., Rosing, K.E., & Storrier, A.L.G. (1996). Applying the flow-capturing location-allocation model to an authentic network: Edmonton, Canada, European Journal of Operational Research (90 pp. 427-443). London, England: Elsevier Science Ltd.

Hodgson, M. J., Rosing, K.E., & Storrier, A.L.G. (1997). Testing a bicriterion location-allocation model with real-world network traffic: the case of Edmonton, Canada. In Climaco, J. (Ed.), Multicriteria Analysis, (pp. 484-93). New York, New York: Springer.

Hodgson, M. J., & Storrier, A.L.G. (1995). Functional error in geographical analysis: the case of spatial separation in location-allocation, Geographical Systems (2 pp. 59-82). Amsterdam: Overseas Publishers Association.

Horn, M. (1996). Analysis and computational schemes for *p*-median heuristics, Environment and Planning A. (28 pp. 1699-1708). London, England: Pion.

Hoyt, H. (1939). The Structure and Growth of Residential Neighbourhoods. Washington, D.C.: Federal Housing Administration.

Lea, A.C. (1978). A model taxonomy and a view of research frontiers in normative locational modelling. International Symposium on Locational Decisions, Banff, Alberta, pp. 166.

Mayhew, S. (1997). Oxford Dictionary of Geography. Oxford, New York: Oxford University Press, pp. 75.

Miller, H. J. (1996). GIS and geometric representation in facility location problems, International Journal of Geographic Information Systems, (10 (7) pp. 791-816). London, England: Taylor & Francis.

Miller, H. J., & Storm, J. D. (1996). Geographic Information System Design for Network Equilibrium-based Travel Demand Models, Transportation Research-C, (4 (6) pp. 373-389). London, England: Elsevier Science Ltd.

Muehrcke, P.C. & Muehrcke, J.O. (1992). Map Use: Reading, Analysis, Interpretation. Madison, Wisconson: J.P. Publications.

Murray, A. T., & Gottsegen, J. M. (1997). The influence of data aggregation on the stability of p-median location model solutions, Geographical Analysis. (29 (3) July 1997). Ohio State University Press.

Nambiar, J. M., Gelders, L. F., & Van Wassenhove, L. (1989). Plant location and vehicle routing in the Malaysian rubber smallholder sector: A case study, European Journal of Operational Research, (38 pp. 14-26). Amsterdam, Belgium: North-Holland Publishing Company.

Nord, M. (1998). Poor people on the move: county-to-county migration and the spatial concentration of poverty, Journal of Regional Science, (38 (2) pp. 329-351). Amherst, Massachusetts: Regional Science Research Institute.

Pocock, D., & Hudson R., (1978). Images of the Urban Environment, London, England: Macmillian.

Rosing, K.E. (1997). An empirical investigation of the vertex substitution heuristic: Environment and Planning B: Planning and Design, (24 pp. 59-67). London, England: Pion.

Rosing, K.E., & Hodgson, M. J. (1996). A Systematic Classification of application of Location-Allocation Models, Belgian Journal of Operations Research, Statistics and Computer Science, (36 (2-3) pp. 77-108).

Statistics Canada, (1996). 1996 Digital Boundary File and Digital Cartographic File: 1996 Census Reference Guide, (Catalogue No. 92F0029XDE, 92F0030XDE, 92F0032XDE - 92F0040XDE). Ottawa, Ontario: Statistics Canada.

Teitz, M. B., & Bart, P. (1968). Heuristic Methods for Estimating the Generalized Vertex Median of a Weighted Graph: Operations Research (16 pp. 955-961) Baltimore, Maryland: Operations Research Society of America.

United Nations, Department of Economic and Social Information Statistics Division (1997). Geographical Information Systems for Population Statistics. New York, New York: United Nations Press.

Valeo C., Baetz, B. W., & Tsanis, I. K. (1998). Location of recycling depots with GIS, Journal of Urban Planning and Development. (June 1998, pp. 93-99). New York, New York: American Society of Civil Engineers, Urban Planning and Development Division

Von Thünen, J. H. (1826). Von Thünen's Isolated State: An English Edition of "Der Isolierte Staat". Translated by Carla M. Wartenberg. Elmsford, New York: Pergamon Press, 1966.

Worboys, M. F. (1997). GIS: A Computing Perspective. London, Britain: Taylor and Francis.

Weber, A., (1909) Theory of the Location of Industires (Translated by C.J. Friedrich, 1929). Chicago, Illinois: University of Chicago Press.

Yates P. M., & Bishop, I. D. (1998). The integration of existing GIS and modelling systems: with urban applications, Computers, Environment and Urban Systems. (22 pp. 71-80).

Appendices

Appendix 1

1996 Census of Population - Attribute database variables

This section lists all of the 1996 Census of Population demographic variables available from the nine Nation Series tables. The Nation Series tables were modified and stored in an external RDBMS for use with the present study. The database code refers to the field name for each demographic variable that is used in the RDBMS for this study. The first digit of the database code refers to the Nation Series table that the data were stored in. The next 3 digits are designed to give each variable a unique number.

Table 1 - Age and Sex, Marital Status/Common Law, Families (Part I: Number, Type and Structure), and Structural Type of Dwelling and Household size

Variables	Database Code
Population, 1996 (100% data)	1001
Total population by sex and age groups (100% data)	1002
Male, total	1003
0-4	1004
5-9	1005
10-14	1006
15	1007
16	1008
17	1009
18	1010
19	1011
15-19	1012
20-24	1013
25-29	1014
30-34	1015
35-39	1016
40-44	1017
45-49	1018
50-54	1019
55-59	1020
60-64	1021
65-69	1022
70-74	1023
75-79	1024
80-84	1025
85+	1026
Female, total	1027
0-4	1028
5-9	1029
10-14	1030
15	1031
16	1032
17	1033

• 18	1034
• 19	1035
• 15-19	1036
• 20-24	1037
• 25-29	1038
• 30-34	1039
• 35-39	1040
• 40-44	1041
• 45-49	1042
• 50-54	1043
• 55-59	1044
• 60-64	1045
• 65-69	1046
• 70-74	1047
• 75-79	1048
• 80-84	1049
• 85+	1050
• Total population 15 years and over by legal marital status (100% data)	1051
• Never married (single)	1052
• Legally married (and not separated)	1053
• Separated, but still legally married	1054
• Divorced	1055
• Widowed	1056
• Total number of census families in private households by family size (20% sample data)	1057
• Size of census family, 2 persons	1058
• Size of census family, 3 persons	1059
• Size of census family, 4 persons	1060
• Size of census family, 5 or more persons	1061
• Total husband-wife families by family structure (20% sample data)	1062
• Total families of now-married couples	1063
• Total without sons and/or daughters at home	1064
• Total with sons and/or daughters at home	1065
• 1 son or daughter	1066
• 2 sons and/or daughters	1067
• 3 or more sons and/or daughters	1068
• Total families of common-law couples	1069
• Total without sons and/or daughters at home	1070
• Total with sons and/or daughters at home	1071
• 1 son or daughter	1072
• 2 sons and/or daughters	1073
• 3 or more sons and/or daughters	1074
• Total lone-parent families by sex of parent (20% sample data)	1075
• Male parent	1076
• 1 son or daughter	1077
• 2 sons and/or daughters	1078
• 3 or more sons and/or daughters	1079
• Female parent	1080
• 1 son or daughter	1081
• 2 sons and/or daughters	1082
• 3 or more sons and/or daughters	1083

Total number of never-married sons and/or daughters at home (20% sample data)	1084
Under 6 years of age	1085
6 - 14 years	1086
15 - 17 years	1087
18 - 24 years	1088
25 years and over	1089
Average number of never-married sons and/or daughters at home per census family	1090
Total number of persons in private households (20% sample data)	1091
Number of non-family persons	1092
Living with relatives	1093
Living with non-relatives only	1094
Living alone	1095
Number of family persons	1096
Average number of persons per census family	1097
Total number of persons 65 years and over (20% sample data)	1098
Number of non-family persons 65 years and over	1099
Living with relatives	1100
Living with non-relatives only	1101
Living alone	1102
Number of family persons 65 years and over	1103
Total number of occupied private dwellings by structural type of dwelling (20% sample data)	1104
Single-detached house	1105
Semi-detached house	1106
Row house	1107
Apartment, detached duplex	1108
Apartment building, five or more storeys	1109
Apartment building, less than five storeys	1110
Other single attached house	1111
Movable dwelling	1112
Total number of private households by household size (20% sample data)	1113
1 person	1114
2 persons	1115
3 persons	1116
4 - 5 persons	1117
6 or more persons	1118

Table 2 - Immigration and Citizenship

Variables	Database Code
Total population by citizenship (20% sample data)	2001
Canadian citizenship	2002
Citizenship other than Canadian	2003
Total population by place of birth (20% sample data)	2004
Non-immigrant population	2005
Born in province of residence	2006
Total immigrants by selected countries of birth	2007

United Kingdom	2008
Italy	2009
United States	2010
Hong Kong	2011
India	2012
China, People's Republic of	2013
Poland	2014
Philippines	2015
Germany	2016
Portugal	2017
Vietnam	2018
Netherlands	2019
Jamaica	2020
Greece	2021
Guyana	2022
Sri Lanka	2023
Lebanon	2024
France	2025
Trinidad and Tobago	2026
Yugoslavia	2027
Hungary	2028
Haiti	2029
Taiwan	2030
Iran	2031
Romania	2032
Korea, South	2033
Ukraine	2034
Pakistan	2035
El Salvador	2036
Egypt	2037
Croatia	2038
Russian Federation	2039
Ireland, Republic of (Eire)	2040
South Africa, Republic of	2041
Mexico	2042
Austria	2043
Chile	2044
Belgium	2045
Fiji	2046
Morocco	2047
Denmark	2048
Czechoslovakia, n.i.e.	2049
Malaysia	2050
Cambodia	2051
Switzerland	2052
Tanzania, United Republic of	2053
Kenya	2054
Iraq	2055
Somalia	2056
Israel	2057
All other places of birth	2058

• Non-permanent residents	2059
• Total recent immigrants by selected countries of birth (20% sample data)	2060
• Hong Kong	2061
• China, People's Republic of	2062
• India	2063
• Philippines	2064
• Sri Lanka	2065
• Poland	2066
• Taiwan	2067
• Vietnam	2068
• United States	2069
• United Kingdom	2070
• Yugoslavia	2071
• Iran	2072
• Jamaica	2073
• Lebanon	2074
• Romania	2075
• Pakistan	2076
• Korea, South	2077
• Guyana	2078
• Trinidad and Tobago	2079
• El Salvador	2080
• Somalia	2081
• France	2082
• Haiti	2083
• Russian Federation	2084
• Iraq	2085
• Portugal	2086
• Bosnia and Herzegovina	2087
• Mexico	2088
• Germany	2089
• Egypt	2090
• Ukraine	2091
• Ghana	2092
• Ethiopia	2093
• South Africa, Republic of	2094
• Bangladesh	2095
• Afghanistan	2096
• Peru	2097
• Guatemala	2098
• Fiji	2099
• Morocco	2100
• All other places of birth	2101
• Total immigrant population by period of immigration (20% sample data)	2102
• Before 1961, period of immigration	2103
• 1961-1970, period of immigration	2104
• 1971-1980, period of immigration	2105
• 1981-1990, period of immigration	2106
• 1991-1996, period of immigration	2107
• Total immigrant population by age at immigration (20% sample data)	2108

0-4 years, age at immigration	2109
5-19 years, age at immigration	2110
20 years and over, age at immigration	2111

Table 3 - Mother Tongue, Home Language

Variables	Database Code
Total population by mother tongue (20% sample data)	3001
Single responses	3002
English	3003
French	3004
Non-official languages	3005
Italian	3006
Chinese	3007
German	3008
Portuguese	3009
Polish	3010
Ukrainian	3011
Spanish	3012
Dutch	3013
Punjabi	3014
Greek	3015
Arabic	3016
Tagalog (Pilipino)	3017
Hungarian	3018
Vietnamese	3019
Cree	3020
Persian (Farsi)	3021
Croatian	3022
Gujarati	3023
Korean	3024
Russian	3025
Hindi	3026
Tamil	3027
Japanese	3028
Creoles	3029
Finnish	3030
Czech	3031
Armenian	3032
Yiddish	3033
Urdu	3034
Inuktitut (Eskimo)	3035
Romanian	3036
Ojibway	3037
Danish	3038
Slovak	3039
Macedonian	3040
Khmer (Cambodian)	3041
Norwegian	3042

Hebrew	3043
Estonian	3044
Swedish	3045
Lao	3046
Lithuanian	3047
Serbian	3048
Latvian (Lettish)	3049
Slovenian	3050
Turkish	3051
Bengali	3052
Maltese	3053
Flemish	3054
Montagnais-Naskapi	3055
Bulgarian	3056
Micmac	3057
Gaelic languages	3058
South Slave	3059
Chipewyan	3060
Dogrib	3061
Kutchin-Gwich'in (Loucheux)	3062
Tlingit	3063
Serbo-Croatian	3064
Dakota/Sioux	3065
Malay-Bahasa	3066
Blackfoot	3067
Malayalam	3068
Thai	3069
Kurdish	3070
Pashto	3071
Other languages	3072
Multiple responses	3073
English and French	3074
English and non-official language	3075
French and non-official language	3076
English, French and non-official language	3077
Total population by knowledge of official languages (20% sample data)	3078
English only	3079
French only	3080
English and French	3081
Neither English nor French	3082
Total population by first official language spoken (20% sample data)	3083
English	3084
French	3085
English and French	3086
Neither English nor French	3087
Official language minority - (number)	3088
Official language minority - (percentage)	3089
Total population by home language (20% sample data)	3090
Single responses	3091
English	3092
French	3093

• Non-official languages	3094
• Chinese	3095
• Italian	3096
• Portuguese	3097
• Spanish	3098
• German	3099
• Polish	3100
• Punjabi	3101
• Greek	3102
• Vietnamese	3103
• Arabic	3104
• Cree	3105
• Tagalog (Pilipino)	3106
• Ukrainian	3107
• Persian (Farsi)	3108
• Korean	3109
• Hungarian	3110
• Tamil	3111
• Gujarati	3112
• Croatian	3113
• Armenian	3114
• Inuktitut (Eskimo)	3115
• Hindi	3116
• Urdu	3117
• Japanese	3118
• Russian	3119
• Creoles	3120
• Dutch	3121
• Khmer (Cambodian)	3122
• Ojibway	3123
• Romanian	3124
• Czech	3125
• Lao	3126
• Macedonian	3127
• Finnish	3128
• Montagnais-Naskapi	3129
• Hebrew	3130
• Yiddish	3131
• Serbian	3132
• Bengali	3133
• Slovak	3134
• Estonian	3135
• Turkish	3136
• Lithuanian	3137
• Latvian (Lettish)	3138
• Micmac	3139
• Slovenian	3140
• Bulgarian	3141
• Serbo-Croatian	3142
• Dakota/Sioux	3143
• South Slave	3144

Malay-Bahasa	3145
Maltese	3146
Blackfoot	3147
Dogrib	3148
Danish	3149
Swedish	3150
Malayalam	3151
Thai	3152
Kurdish	3153
Pashto	3154
Flemish	3155
Chipewyan	3156
Kutchin-Gwich'in (Loucheux)	3157
Norwegian	3158
Gaelic languages	3159
Tlingit	3160
Other languages	3161
Multiple responses	3162
English and French	3163
English and non-official language	3164
French and non-official language	3165
English, French and non-official language	3166
Knowledge of non-official languages (20% sample data): Italian	3167
German	3168
Chinese	3169
Spanish	3170
Portuguese	3171
Ukrainian	3172
Polish	3173
Dutch	3174
Punjabi	3175
Arabic	3176
Greek	3177
Tagalog (Pilipino)	3178
Vietnamese	3179
Hindi	3180
Hungarian	3181
Cree	3182
Russian	3183
Gujarati	3184
Yiddish	3185
Hebrew	3186
Urdu	3187
Creoles	3188
Persian (Farsi)	3189
Croatian	3190
Japanese	3191
Korean	3192
Tamil	3193
Finnish	3194
Armenian	3195

Romanian	3196
Ojibway	3197
Czech	3198
Danish	3199
Non-verbal languages	3200
Inuktitut (Eskimo)	3201
Turkish	3202
Macedonian	3203
Slovak	3204
Khmer (Cambodian)	3205
Swedish	3206
Swahili	3207
Norwegian	3208
Lao	3209
Malay-Bahasa	3210
Serbian	3211
Lithuanian	3212
Estonian	3213
Latvian (Lettish)	3214
Bengali	3215
Maltese	3216
Flemish	3217
Slovenian	3218
Gaelic languages	3219
Sinhalese	3220
Montagnais-Naskapi	3221
Serbo-Croatian	3222
Thai	3223
Micmac	3224
Blackfoot	3225
Bulgarian	3226
Malayalam	3227
Dakota/Sioux	3228
South Slave	3229
Icelandic	3230
Nishga	3231
Frisian	3232
Chipewyan	3233
Kutchin-Gwich'in (Loucheux)	3234
Tlingit	3235
Dogrib	3236
Kurdish	3237
Pashto	3238
Other languages	3239

Table 4 - Aboriginal

Variables	Database Code
Total population by Aboriginal groups and non-Aboriginal population	

(20% sample data)	4001
Total Aboriginal population	4002
North American Indian single response	4003
Metis single response	4004
Inuit single response	4005
Multiple Aboriginal responses	4006
Other Aboriginal response	4007
Total non-Aboriginal population	4008

Table 5 - Ethnic Origin, Visible Minorities

Variables	Database Code
Total population by ethnic origin (single and multiple responses) (20% sample data)	5001
Total population - Single responses	5002
Total population - Multiple responses	5003
Canadian - Total responses	5004
Canadian - Single responses	5005
Canadian - Multiple responses	5006
French - Total responses	5007
French - Single responses	5008
French - Multiple responses	5009
English - Total responses	5010
English - Single responses	5011
English - Multiple responses	5012
Chinese - Total responses	5013
Chinese - Single responses	5014
Chinese - Multiple responses	5015
Italian - Total responses	5016
Italian - Single responses	5017
Italian - Multiple responses	5018
German - Total responses	5019
German - Single responses	5020
German - Multiple responses	5021
Scottish - Total responses	5022
Scottish - Single responses	5023
Scottish - Multiple responses	5024
Irish - Total responses	5025
Irish - Single responses	5026
Irish - Multiple responses	5027
East Indian - Total responses	5028
East Indian - Single responses	5029
East Indian - Multiple responses	5030
North American Indian - Total responses	5031
North American Indian - Single responses	5032
North American Indian - Multiple responses	5033
Ukrainian - Total responses	5034
Ukrainian - Single responses	5035

Ukrainian - Multiple responses	5036
Dutch (Netherlands) - Total responses	5037
Dutch (Netherlands) - Single responses	5038
Dutch (Netherlands) - Multiple responses	5039
Polish - Total responses	5040
Polish - Single responses	5041
Polish - Multiple responses	5042
Portuguese - Total responses	5043
Portuguese - Single responses	5044
Portuguese - Multiple responses	5045
Filipino - Total responses	5046
Filipino - Single responses	5047
Filipino - Multiple responses	5048
Jewish - Total responses	5049
Jewish - Single responses	5050
Jewish - Multiple responses	5051
Greek - Total responses	5052
Greek - Single responses	5053
Greek - Multiple responses	5054
Jamaican - Total responses	5055
Jamaican - Single responses	5056
Jamaican - Multiple responses	5057
Vietnamese - Total responses	5058
Vietnamese - Single responses	5059
Vietnamese - Multiple responses	5060
Hungarian (Magyar) - Total responses	5061
Hungarian (Magyar) - Single responses	5062
Hungarian (Magyar) - Multiple responses	5063
Lebanese - Total responses	5064
Lebanese - Single responses	5065
Lebanese - Multiple responses	5066
Spanish - Total responses	5067
Spanish - Single responses	5068
Spanish - Multiple responses	5069
Haitian - Total responses	5070
Haitian - Single responses	5071
Haitian - Multiple responses	5072
Korean - Total responses	5073
Korean - Single responses	5074
Korean - Multiple responses	5075
Quebecois - Total responses	5076
Quebecois - Single responses	5077
Quebecois - Multiple responses	5078
Croatian - Total responses	5079
Croatian - Single responses	5080
Croatian - Multiple responses	5081
Iranian - Total responses	5082
Iranian - Single responses	5083
Iranian - Multiple responses	5084
Japanese - Total responses	5085
Japanese - Single responses	5086

Japanese - Multiple responses	5087
Metis - Total responses	5088
Metis - Single responses	5089
Metis - Multiple responses	5090
Norwegian - Total responses	5091
Norwegian - Single responses	5092
Norwegian - Multiple responses	5093
Russian - Total responses	5094
Russian - Single responses	5095
Russian - Multiple responses	5096
British, n.i.e. - Total responses	5097
British, n.i.e. - Single responses	5098
British, n.i.e. - Multiple responses	5099
Romanian - Total responses	5100
Romanian - Single responses	5101
Romanian - Multiple responses	5102
Danish - Total responses	5103
Danish - Single responses	5104
Danish - Multiple responses	5105
Finnish - Total responses	5106
Finnish - Single responses	5107
Finnish - Multiple responses	5108
Inuit - Total responses	5109
Inuit - Single responses	5110
Inuit - Multiple responses	5111
Arab, n.i.e. - Total responses	5112
Arab, n.i.e. - Single responses	5113
Arab, n.i.e. - Multiple responses	5114
Sri Lankan - Total responses	5115
Sri Lankan - Single responses	5116
Sri Lankan - Multiple responses	5117
Belgian - Total responses	5118
Belgian - Single responses	5119
Belgian - Multiple responses	5120
West Indian - Total responses	5121
West Indian - Single responses	5122
West Indian - Multiple responses	5123
Swedish - Total responses	5124
Swedish - Single responses	5125
Swedish - Multiple responses	5126
Punjabi - Total responses	5127
Punjabi - Single responses	5128
Punjabi - Multiple responses	5129
African (Black), n.i.e. - Total responses	5130
African (Black), n.i.e. - Single responses	5131
African (Black), n.i.e. - Multiple responses	5132
Serbian - Total responses	5133
Serbian - Single responses	5134
Serbian - Multiple responses	5135
Somali - Total responses	5136
Somali - Single responses	5137

• Somali - Multiple responses	5138
• Austrian - Total responses	5139
• Austrian - Single responses	5140
• Austrian - Multiple responses	5141
• Yugoslav, n.i.e. - Total responses	5142
• Yugoslav, n.i.e. - Single responses	5143
• Yugoslav, n.i.e. - Multiple responses	5144
• Black - Total responses	5145
• Black - Single responses	5146
• Black - Multiple responses	5147
• Welsh - Total responses	5148
• Welsh - Single responses	5149
• Welsh - Multiple responses	5150
• Pakistani - Total responses	5151
• Pakistani - Single responses	5152
• Pakistani - Multiple responses	5153
• South Asian, n.i.e. - Total responses	5154
• South Asian, n.i.e. - Single responses	5155
• South Asian, n.i.e. - Multiple responses	5156
• Armenian - Total responses	5157
• Armenian - Single responses	5158
• Armenian - Multiple responses	5159
• Czech - Total responses	5160
• Czech - Single responses	5161
• Czech - Multiple responses	5162
• Egyptian - Total responses	5163
• Egyptian - Single responses	5164
• Egyptian - Multiple responses	5165
• Chilean - Total responses	5166
• Chilean - Single responses	5167
• Chilean - Multiple responses	5168
• Swiss - Total responses	5169
• Swiss - Single responses	5170
• Swiss - Multiple responses	5171
• Trinidadian/Tobagonian - Total responses	5172
• Trinidadian/Tobagonian - Single responses	5173
• Trinidadian/Tobagonian - Multiple responses	5174
• Guyanese - Total responses	5175
• Guyanese - Single responses	5176
• Guyanese - Multiple responses	5177
• American - Total responses	5178
• American - Single responses	5179
• American - Multiple responses	5180
• Latin/Central/South American, n.i.e. - Total responses	5181
• Latin/Central/South American, n.i.e. - Single responses	5182
• Latin/Central/South American, n.i.e. - Multiple responses	5183
• Slovak - Total responses	5184
• Slovak - Single responses	5185
• Slovak - Multiple responses	5186
• Salvadorean - Total responses	5187
• Salvadorean - Single responses	5188

- Salvadorean - Multiple responses 5189
- Macedonian - Total responses 5190
- Macedonian - Single responses 5191
- Macedonian - Multiple responses 5192
- Acadian - Total responses 5193
- Acadian - Single responses 5194
- Acadian - Multiple responses 5195
- Tamil - Total responses 5196
- Tamil - Single responses 5197
- Tamil - Multiple responses 5198
- Cambodian - Total responses 5199
- Cambodian - Single responses 5200
- Cambodian - Multiple responses 5201
- Slovenian - Total responses 5202
- Slovenian - Single responses 5203
- Slovenian - Multiple responses 5204
- Maltese - Total responses 5205
- Maltese - Single responses 5206
- Maltese - Multiple responses 5207
- Laotian - Total responses 5208
- Laotian - Single responses 5209
- Laotian - Multiple responses 5210
- Ethiopian - Total responses 5211
- Ethiopian - Single responses 5212
- Ethiopian - Multiple responses 5213
- Ghanaian - Total responses 5214
- Ghanaian - Single responses 5215
- Ghanaian - Multiple responses 5216
- Lithuanian - Total responses 5217
- Lithuanian - Single responses 5218
- Lithuanian - Multiple responses 5219
- Afghan - Total responses 5220
- Afghan - Single responses 5221
- Afghan - Multiple responses 5222
- Icelandic - Total responses 5223
- Icelandic - Single responses 5224
- Icelandic - Multiple responses 5225
- Estonian - Total responses 5226
- Estonian - Single responses 5227
- Estonian - Multiple responses 5228
- Turk - Total responses 5229
- Turk - Single responses 5230
- Turk - Multiple responses 5231
- Latvian - Total responses 5232
- Latvian - Single responses 5233
- Latvian - Multiple responses 5234
- Barbadian - Total responses 5235
- Barbadian - Single responses 5236
- Barbadian - Multiple responses 5237
- Syrian - Total responses 5238
- Syrian - Single responses 5239

• Syrian - Multiple responses	5240
• Czechoslovakian - Total responses	5241
• Czechoslovakian - Single responses	5242
• Czechoslovakian - Multiple responses	5243
• Mexican - Total responses	5244
• Mexican - Single responses	5245
• Mexican - Multiple responses	5246
• Peruvian - Total responses	5247
• Peruvian - Single responses	5248
• Peruvian - Multiple responses	5249
• Moroccan - Total responses	5250
• Moroccan - Single responses	5251
• Moroccan - Multiple responses	5252
• Iraqi - Total responses	5253
• Iraqi - Single responses	5254
• Iraqi - Multiple responses	5255
• Bosnian - Total responses	5256
• Bosnian - Single responses	5257
• Bosnian - Multiple responses	5258
• Palestinian - Total responses	5259
• Palestinian - Single responses	5260
• Palestinian - Multiple responses	5261
• Scandinavian, n.i.e. - Total response	5262
• Scandinavian, n.i.e. - Single responses	5263
• Scandinavian, n.i.e. - Multiple responses	5264
• Caribbean, n.i.e. - Total responses	5265
• Caribbean, n.i.e. - Single responses	5266
• Caribbean, n.i.e. - Multiple responses	5267
• European, n.i.e. - Total responses	5268
• European, n.i.e. - Single responses	5269
• European, n.i.e. - Multiple responses	5270
• Bulgarian - Total responses	5271
• Bulgarian - Single responses	5272
• Bulgarian - Multiple responses	5273
• Guatemalan - Total responses	5274
• Guatemalan - Single responses	5275
• Guatemalan - Multiple responses	5276
• Bangladeshi - Total responses	5277
• Bangladeshi - Single responses	5278
• Bangladeshi - Multiple responses	5279
• Taiwanese - Total responses	5280
• Taiwanese - Single responses	5281
• Taiwanese - Multiple responses	5282
• Colombian - Total responses	5283
• Colombian - Single responses	5284
• Colombian - Multiple responses	5285
• Eritrean - Total responses	5286
• Eritrean - Single responses	5287
• Eritrean - Multiple responses	5288
• Fijian - Total responses	5289
• Fijian - Single responses	5290

Fijian - Multiple responses	5291
Ecuadorian - Total responses	5292
Ecuadorian - Single responses	5293
Ecuadorian - Multiple responses	5294
Algerian - Total responses	5295
Algerian - Single responses	5296
Algerian - Multiple responses	5297
Nigerian - Total responses	5298
Nigerian - Single responses	5299
Nigerian - Multiple responses	5300
Hispanic - Total responses	5301
Hispanic - Single responses	5302
Hispanic - Multiple responses	5303
Total - Total population by visible minority population	5304
Total visible minority population	5305
Black	5306
South Asian	5307
Chinese	5308
Korean	5309
Japanese	5310
Southeast Asian	5311
Filipino	5312
Arab/West Asian	5313
Latin American	5314
Visible minority, n.i.e.	5315
Multiple visible minority	5316
All others	5317

Table 6 - Labour Force Activities, Occupation and Class of Worker, Household Activities, Place of Work, Mode of Transportation to Work, Commuting Distance

Variables	Database Code
Total population 15 years and over by labour force activity (20% sample data)	6001
In the labour force	6002
Employed	6003
Unemployed	6004
Not in the labour force	6005
Participation rate	6006
Employment-population ratio	6007
Unemployment rate	6008
Total population 15-24 years by labour force activity (20% sample data)	6009
In the labour force	6010
Employed	6011
Unemployed	6012
Not in the labour force	6013
Participation rate	6014
Employment-population ratio	6015
Unemployment rate	6016

• Total population 25 years and over by labour force activity (20% sample data)	6017
• In the labour force	6018
• Employed	6019
• Unemployed	6020
• Not in the labour force	6021
• Participation rate	6022
• Employment-population ratio	6023
• Unemployment rate	6024
• Males 15 years and over by labour force activity (20% sample data)	6025
• In the labour force	6026
• Employed	6027
• Unemployed	6028
• Not in the labour force	6029
• Participation rate	6030
• Employment-population ratio	6031
• Unemployment rate	6032
• Males 15-24 years by labour force activity (20% sample data)	6033
• In the labour force	6034
• Employed	6035
• Unemployed	6036
• Not in the labour force	6037
• Participation rate	6038
• Employment-population ratio	6039
• Unemployment rate	6040
• Males 25 years and over by labour force activity (20% sample data)	6041
• In the labour force	6042
• Employed	6043
• Unemployed	6044
• Not in the labour force	6045
• Participation rate	6046
• Employment-population ratio	6047
• Unemployment rate	6048
• Females 15 years and over by labour force activity (20% sample data)	6049
• In the labour force	6050
• Employed	6051
• Unemployed	6052
• Not in the labour force	6053
• Participation rate	6054
• Employment-population ratio	6055
• Unemployment rate	6056
• Females 15-24 years by labour force activity (20% sample data)	6057
• In the labour force	6058
• Employed	6059
• Unemployed	6060
• Not in the labour force	6061
• Participation rate	6062
• Employment-population ratio	6063
• Unemployment rate	6064
• Females 25 years and over by labour force activity (20% sample data)	6065
• In the labour force	6066

Employed	6067
Unemployed	6068
Not in the labour force	6069
Participation rate	6070
Employment-population ratio	6071
Unemployment rate	6072
Population 15 years + in households by presence of children/labour force activity (20% data)	6073
In the labour force	6074
Employed	6075
Unemployed	6076
Not in the labour force	6077
Participation rate	6078
Employment-population ratio	6079
Unemployment rate	6080
Population 15 years and over in private households with no children at home	6081
In the labour force	6082
Employed	6083
Unemployed	6084
Not in the labour force	6085
Participation rate	6086
Employment-population ratio	6087
Unemployment rate	6088
Population 15 years and over in private households with children at home	6089
In the labour force	6090
Employed	6091
Unemployed	6092
Not in the labour force	6093
Participation rate	6094
Employment-population ratio	6095
Unemployment rate	6096
Population 15 years and over in private households with children under 6 years only	6097
In the labour force	6098
Employed	6099
Unemployed	6100
Not in the labour force	6101
Participation rate	6102
Employment-population ratio	6103
Unemployment rate	6104
Population 15 years + in private households with children under 6 years and children 6 years and over	6105
In the labour force	6106
Employed	6107
Unemployed	6108
Not in the labour force	6109
Participation rate	6110
Employment-population ratio	6111
Unemployment rate	6112
Population 15 years and over in private households with children	

6 years and over only	6113
• In the labour force	6114
• Employed	6115
• Unemployed	6116
• Not in the labour force	6117
• Participation rate	6118
• Employment-population ratio	6119
• Unemployment rate	6120
• Males 15 years + in private households by presence of children and labour force activity (20% data)	6121
• In the labour force	6122
• Employed	6123
• Unemployed	6124
• Not in the labour force	6125
• Participation rate	6126
• Employment-population ratio	6127
• Unemployment rate	6128
• Males 15 years and over in private households with no children at home	6129
• In the labour force	6130
• Employed	6131
• Unemployed	6132
• Not in the labour force	6133
• Participation rate	6134
• Employment-population ratio	6135
• Unemployment rate	6136
• Males 15 years and over in private households with children at home	6137
• In the labour force	6138
• Employed	6139
• Unemployed	6140
• Not in the labour force	6141
• Participation rate	6142
• Employment-population ratio	6143
• Unemployment rate	6144
• Males 15 years and over in private households with children under 6 years only	6145
• In the labour force	6146
• Employed	6147
• Unemployed	6148
• Not in the labour force	6149
• Participation rate	6150
• Employment-population ratio	6151
• Unemployment rate	6152
• Males 15 years and over in private households with children under 6 years and children 6 years and over	6153
• In the labour force	6154
• Employed	6155
• Unemployed	6156
• Not in the labour force	6157
• Participation rate	6158
• Employment-population ratio	6159
• Unemployment rate	6160

Males 15 years and over in private households with children 6 years and over only	6161
In the labour force	6162
Employed	6163
Unemployed	6164
Not in the labour force	6165
Participation rate	6166
Employment-population ratio	6167
Unemployment rate	6168
Females 15 years + in private households by presence of children and labour force activity (20% data)	6169
In the labour force	6170
Employed	6171
Unemployed	6172
Not in the labour force	6173
Participation rate	6174
Employment-population ratio	6175
Unemployment rate	6176
Females 15 years and over in private households with no children at home	6177
In the labour force	6178
Employed	6179
Unemployed	6180
Not in the labour force	6181
Participation rate	6182
Employment-population ratio	6183
Unemployment rate	6184
Females 15 years and over in private households with children at home	6185
In the labour force	6186
Employed	6187
Unemployed	6188
Not in the labour force	6189
Participation rate	6190
Employment-population ratio	6191
Unemployment rate	6192
Females 15 years and over in private households with children under 6 years only	6193
In the labour force	6194
Employed	6195
Unemployed	6196
Not in the labour force	6197
Participation rate	6198
Employment-population ratio	6199
Unemployment rate	6200
Females 15 years and over in private households with children under 6 years and children 6 years and over	6201
In the labour force	6202
Employed	6203
Unemployed	6204
Not in the labour force	6205
Participation rate	6206
Employment-population ratio	6207

Unemployment rate	6208
Females 15 years and over in private households with children 6 years and over only	6209
In the labour force	6210
Employed	6211
Unemployed	6212
Not in the labour force	6213
Participation rate	6214
Employment-population ratio	6215
Unemployment rate	6216
Total labour force 15 years and over by industry divisions (20% sample data)	6217
Industry - Not applicable	6218
All industries	6219
Division A - Agricultural and related service industries	6220
Division B - Fishing and trapping industries	6221
Division C - Logging and forestry industries	6222
Division D - Mining (including milling), quarrying and oil well industries	6223
Division E - Manufacturing industries	6224
Division F - Construction industries	6225
Division G - Transportation and storage industries	6226
Division H - Communication and other utility industries	6227
Division I - Wholesale trade industries	6228
Division J - Retail trade industries	6229
Division K - Finance and insurance industries	6230
Division L - Real estate operator and insurance agent industries	6231
Division M - Business service industries	6232
Division N - Government service industries	6233
Division O - Educational service industries	6234
Division P - Health and social service industries	6235
Division Q - Accommodation, food and beverage service industries	6236
Division R - Other service industries	6237
Total labour force 15 years + by occupational broad categories and occupation major groups (20% data)	6238
Occupation - Not applicable	6239
All occupations	6240
A Management occupations	6241
A0 Senior management occupations	6242
A1 Specialist managers	6243
A2 Managers in retail trade, food and accommodation services	6244
A3 Other managers n.e.c.	6245
B Business, finance and administrative occupations	6246
B0 Professional occupations in business and finance	6247
B1 Finance and insurance administrative occupations	6248
B2 Secretaries	6249
B3 Administrative and regulatory occupations	6250
B4 Clerical supervisors	6251
B5 Clerical occupations	6252
C Natural and applied sciences and related occupations	6253
C0 Professional occupations in natural and applied sciences	6254

- C1 Technical occupations related to natural and applied sciences 6255
- D Health occupations 6256
- D0 Professional occupations in health 6257
- D1 Nurse supervisors and registered nurses 6258
- D2 Technical and related occupations in health 6259
- D3 Assisting occupations in support of health services 6260
- E Occupations in social science, education, government service and religion 6261
- E0 Judges, lawyers, psychologists, social workers, ministers of religion, and policy and program officers 6262
- E1 Teachers and professors 6263
- E2 Paralegals, social services workers and occupations in education and religion n.e.c. 6264
- F Occupations in art, culture, recreation and sport 6265
- F0 Professional occupations in art and culture 6266
- F1 Technical occupations in art, culture, recreation and sport 6267
- G Sales and service occupations 6268
- G0 Sales and service supervisors 6269
- G1 Wholesale, technical, insurance, real estate sales specialists, and retail, wholesale and grain buyers 6270
- G2 Retail salespersons and sales clerks 6271
- G3 Cashiers 6272
- G4 Chefs and cooks 6273
- G5 Occupations in food and beverage service 6274
- G6 Occupations in protective services 6275
- G7 Occupations in travel and accommodation including attendants in recreation and sport 6276
- G8 Childcare and home support workers 6277
- G9 Sales and service occupations n.e.c. 6278
- H Trades, transport and equipment operators and related occupations 6279
- H0 Contractors and supervisors in trades and transportation 6280
- H1 Construction trades 6281
- H2 Stationary engineers, power station operators and electrical trades and telecommunications occupations 6282
- H3 Machinists, metal forming, shaping and erecting occupations 6283
- H4 Mechanics 6284
- H5 Other trades n.e.c. 6285
- H6 Heavy equipment and crane operators including drillers 6286
- H7 Transportation equipment operators and related workers, excluding labourers 6287
- H8 Trades helpers, construction, and transportation labourers and related occupations 6288
- I Occupations unique to primary industry 6289
- I0 Occupations unique to agriculture excluding labourers 6290
- I1 Occupations unique to forestry operations, mining, oil and gas extraction, and fishing, excluding labourers 6291
- I2 Primary production labourers 6292
- J Occupations unique to processing, manufacturing and utilities 6293
- J0 Supervisors in manufacturing 6294
- J1 Machine operators in manufacturing 6295
- J2 Assemblers in manufacturing 6296

J3 Labourers in processing, manufacturing and utilities	6297
Male labour force 15 years + by occupational broad categories and occupation major groups (20% data)	6298
Occupation - Not applicable	6299
All occupations	6300
A Management occupations	6301
A0 Senior management occupations	6302
A1 Specialist managers	6303
A2 Managers in retail trade, food and accommodation services	6304
A3 Other managers n.e.c.	6305
B Business, finance and administrative occupations	6306
B0 Professional occupations in business and finance	6307
B1 Finance and insurance administrative occupations	6308
B2 Secretaries	6309
B3 Administrative and regulatory occupations	6310
B4 Clerical supervisors	6311
B5 Clerical occupations	6312
C Natural and applied sciences and related occupations	6313
C0 Professional occupations in natural and applied sciences	6314
C1 Technical occupations related to natural and applied sciences	6315
D Health occupations	6316
D0 Professional occupations in health	6317
D1 Nurse supervisors and registered nurses	6318
D2 Technical and related occupations in health	6319
D3 Assisting occupations in support of health services	6320
E Occupations in social science, education, government service and religion	6321
E0 Judges, lawyers, psychologists, social workers, ministers of religion, and policy and program officers	6322
E1 Teachers and professors	6323
E2 Paralegals, social services workers and occupations in education and religion n.e.c.	6324
F Occupations in art, culture, recreation and sport	6325
F0 Professional occupations in art and culture	6326
F1 Technical occupations in art, culture, recreation and sport	6327
G Sales and service occupations	6328
G0 Sales and service supervisors	6329
G1 Wholesale, technical, insurance, real estate sales specialists, and retail, wholesale and grain buyers	6330
G2 Retail salespersons and sales clerks	6331
G3 Cashiers	6332
G4 Chefs and cooks	6333
G5 Occupations in food and beverage service	6334
G6 Occupations in protective services	6335
G7 Occupations in travel and accommodation including attendants in recreation and sport	6336
G8 Childcare and home support workers	6337
G9 Sales and service occupations n.e.c.	6338
H Trades, transport and equipment operators and related occupations	6339
H0 Contractors and supervisors in trades and transportation	6340
H1 Construction trades	6341

H2 Stationary engineers, power station operators and electrical trades and telecommunications occupations	6342
H3 Machinists, metal forming, shaping and erecting occupations	6343
H4 Mechanics	6344
H5 Other trades n.e.c.	6345
H6 Heavy equipment and crane operators including drillers	6346
H7 Transportation equipment operators and related workers, excluding labourers	6347
H8 Trades helpers, construction, and transportation labourers and related occupations	6348
I Occupations unique to primary industry	6349
I0 Occupations unique to agriculture excluding labourers	6350
I1 Occupations unique to forestry operations, mining, oil and gas extraction, and fishing, excluding labourers	6351
I2 Primary production labourers	6352
J Occupations unique to processing, manufacturing and utilities	6353
J0 Supervisors in manufacturing	6354
J1 Machine operators in manufacturing	6355
J2 Assemblers in manufacturing	6356
J3 Labourers in processing, manufacturing and utilities	6357
Female labour force 15 years + by occupational broad categories and occupation major groups (20% data)	6358
Occupation - Not applicable	6359
All occupations	6360
A Management occupations	6361
A0 Senior management occupations	6362
A1 Specialist managers	6363
A2 Managers in retail trade, food and accommodation services	6364
A3 Other managers n.e.c.	6365
B Business, finance and administrative occupations	6366
B0 Professional occupations in business and finance	6367
B1 Finance and insurance administrative occupations	6368
B2 Secretaries	6369
B3 Administrative and regulatory occupations	6370
B4 Clerical supervisors	6371
B5 Clerical occupations	6372
C Natural and applied sciences and related occupations	6373
C0 Professional occupations in natural and applied sciences	6374
C1 Technical occupations related to natural and applied sciences	6375
D Health occupations	6376
D0 Professional occupations in health	6377
D1 Nurse supervisors and registered nurses	6378
D2 Technical and related occupations in health	6379
D3 Assisting occupations in support of health services	6380
E Occupations in social science, education, government service and religion	6381
E0 Judges, lawyers, psychologists, social workers, ministers of religion, and policy and program officers	382
E1 Teachers and professors	6383
E2 Paralegals, social services workers and occupations in education and religion n.e.c.	6384

F Occupations in art, culture, recreation and sport	6385
F0 Professional occupations in art and culture	6386
F1 Technical occupations in art, culture, recreation and sport	6387
G Sales and service occupations	6388
G0 Sales and service supervisors	6389
G1 Wholesale, technical, insurance, real estate sales specialists, and retail, wholesale and grain buyers	6390
G2 Retail salespersons and sales clerks	6391
G3 Cashiers	6392
G4 Chefs and cooks	6393
G5 Occupations in food and beverage service	6394
G6 Occupations in protective services	6395
G7 Occupations in travel and accommodation including attendants in recreation and sport	6396
G8 Childcare and home support workers	6397
G9 Sales and service occupations n.e.c.	6398
H Trades, transport and equipment operators and related occupations	6399
H0 Contractors and supervisors in trades and transportation	6400
H1 Construction trades	6401
H2 Stationary engineers, power station operators and electrical trades and telecommunications occupations	6402
H3 Machinists, metal forming, shaping and erecting occupations	6403
H4 Mechanics	6404
H5 Other trades n.e.c.	6405
H6 Heavy equipment and crane operators including drillers	6406
H7 Transportation equipment operators and related workers, excluding labourers	6407
H8 Trades helpers, construction, and transportation labourers and related occupations	6408
I Occupations unique to primary industry	6409
I0 Occupations unique to agriculture excluding labourers	6410
I1 Occupations unique to forestry operations, mining, oil and gas extraction, and fishing, excluding labourers	6411
I2 Primary production labourers	6412
J Occupations unique to processing, manufacturing and utilities	6413
J0 Supervisors in manufacturing	6414
J1 Machine operators in manufacturing	6415
J2 Assemblers in manufacturing	6416
J3 Labourers in processing, manufacturing and utilities	6417
Total labour force 15 years and over by class of worker (20% sample data)	6418
Class of worker - Not applicable	6419
All classes of worker	6420
Paid workers	6421
Employees	6422
Self-employed (incorporated)	6423
Without paid help	6424
With paid help	6425
Self-employed (unincorporated)	6426
Without paid help	6427
With paid help	6428
Unpaid family workers	6429

Male labour force 15 years and over by class of worker (20% sample data)	6430
Class of worker - Not applicable	6431
All classes of worker	6432
Paid workers	6433
Employees	6434
Self-employed (incorporated)	6435
Without paid help	6436
With paid help	6437
Self-employed (unincorporated)	6438
Without paid help	6439
With paid help	6440
Unpaid family workers	6441
Female labour force 15 years and over by class of worker (20% sample data)	6442
Class of worker - Not applicable	6443
All classes of worker	6444
Paid workers	6445
Employees	6446
Self-employed (incorporated)	6447
Without paid help	6448
With paid help	6449
Self-employed (unincorporated)	6450
Without paid help	6451
With paid help	6452
Unpaid family workers	6453
Population 15 years and over by hours of unpaid housework (20% sample data)	6454
No hours of housework	6455
Less than 5 hours of housework	6456
5 to 14 hours of housework	6457
15 to 29 hours of housework	6458
30 to 59 hours of housework	6459
60 or more hours of housework	6460
Males 15 years and over by hours of unpaid housework (20% sample data)	6461
No hours of housework	6462
Less than 5 hours of housework	6463
5 to 14 hours of housework	6464
15 to 29 hours of housework	6465
30 to 59 hours of housework	6466
60 or more hours of housework	6467
Females 15 years and over by hours of unpaid housework (20% sample data)	6468
No hours of housework	6469
Less than 5 hours of housework	6470
5 to 14 hours of housework	6471
15 to 29 hours of housework	6472
30 to 59 hours of housework	6473
60 or more hours of housework	6474
Population 15 years and over by hours of unpaid childcare (20% sample data)	6475
No hours of childcare	6476
Less than 5 hours of childcare	6477

5 to 14 hours of childcare	6478
15 to 29 hours of childcare	6479
30 to 59 hours of childcare	6480
60 or more hours of childcare	6481
Males 15 years and over by hours of unpaid childcare (20% sample data)	6482
No hours of childcare	6483
Less than 5 hours of childcare	6484
5 to 14 hours of childcare	6485
15 to 29 hours of childcare	6486
30 to 59 hours of childcare	6487
60 or more hours of childcare	6488
Females 15 years and over by hours of unpaid childcare (20% sample data)	6489
No hours of childcare	6490
Less than 5 hours of childcare	6491
5 to 14 hours of childcare	6492
15 to 29 hours of childcare	6493
30 to 59 hours of childcare	6494
60 or more hours of childcare	6495
Population 15 years and over by hours of unpaid care to seniors (20% sample data)	6496
No hours of care to seniors	6497
Less than 5 hours of care to seniors	6498
5 to 9 hours of care to seniors	6499
10 or more hours of care to seniors	6500
Males 15 years and over by hours of unpaid care to seniors (20% sample data)	6501
No hours of care to seniors	6502
Less than 5 hours of care to seniors	6503
5 to 9 hours of care to seniors	6504
10 or more hours of care to seniors	6505
Females 15 years and over by hours of unpaid care to seniors (20% sample data)	6506
No hours of care to seniors	6507
Less than 5 hours of care to seniors	6508
5 to 9 hours of care to seniors	6509
10 or more hours of care to seniors	6510
Total employed labour force 15 years and over by place of work status (20% sample data)	6511
Males	6512
Usual place of work	6513
At home	6514
Outside Canada	6515
No fixed workplace	6516
Females	6517
Usual place of work	6518
At home	6519
Outside Canada	6520
No fixed workplace	6521
Total employed labour force 15 years and over by mode of transportation (20% sample data)	6522
Males with usual place of work or no fixed workplace	6523

• Car, truck, van as driver	6524
• Car, truck, van as passenger	6525
• Public transit	6526
• Walked to work	6527
• Bicycle	6528
• Motorcycle	6529
• Taxicab	6530
• Other method	6531
• Females with usual place of work or no fixed workplace	6532
• Car, truck, van as driver	6533
• Car, truck, van as passenger	6534
• Public transit	6535
• Walked to work	6536
• Bicycle	6537
• Motorcycle	6538
• Taxicab	6539

Table 7 - Education, Mobility and Migration

Variables	Database Code
• Total population, 15 to 24 years by school attendance (20% sample data)	7001
• Not attending school	7002
• Attending school full-time	7003
• Attending school part-time	7004
• Total population 15 years and over by highest level of schooling (20% sample data)	7005
• Less than grade 9	7006
• Grades 9 to 13	7007
• Without secondary school graduation certificate	7008
• With secondary school graduation certificate	7009
• Trades certificate or diploma	7010
• Other non-university education only	7011
• Without certificate or diploma	7012
• With certificate or diploma	7013
• University	7014
• Without degree	7015
• Without certificate or diploma	7016
• With certificate or diploma	7017
• With bachelor's degree or higher	7018
• Males with post secondary qualifications by major field of study (20% sample data)	7019
• Educational, recreational and counselling services	7020
• Fine and applied arts	7021
• Humanities and related fields	7022
• Social sciences and related fields	7023
• Commerce, management and business administration	7024
• Agricultural and biological sciences/technologies	7025
• Engineering and applied sciences	7026
• Engineering and applied science technologies and trades	7027

• Health professions, sciences and technologies	7028
• Mathematics and physical sciences	7029
• No specialization and all other, n.e.c.	7030
• Females with post secondary qualifications by major field of study (20% sample data)	7031
• Educational, recreational and counselling services	7032
• Fine and applied arts	7033
• Humanities and related fields	7034
• Social sciences and related fields	7035
• Commerce, management and business administration	7036
• Agricultural and biological sciences/technologies	7037
• Engineering and applied sciences	7038
• Engineering and applied science technologies and trades	7039
• Health professions, sciences and technologies	7040
• Mathematics and physical sciences	7041
• No specialization and all other, n.e.c.	7042
• Total by mobility status 1 year ago (20% sample data)	7043
• Non-movers	7044
• Movers	7045
• Non-migrants	7046
• Migrants	7047
• Internal migrants	7048
• Intraprovincial migrants	7049
• Interprovincial migrants	7050
• External migrants	7051
• Total by mobility status 5 years ago (20% sample data)	7052
• Non-movers	7053
• Movers	7054
• Non-migrants	7055
• Migrants	7056
• Internal migrants	7057
• Intraprovincial migrants	7058
• Interprovincial migrants	7059
• External migrants	7060

Table 8 - Sources of Income

Variables	Database Code
• Total income of population 15 years and over (20% sample data)	8001
• Without income	8002
• With income	8003
• Average 1995 income $	8004
• Standard error of average income $	8005
• 15-24 years	8006
• Without income	8007
• With income	8008
• Average 1995 income $	8009
• Standard error of average income $	8010
• 25-44 years	8011

• Without income	8012
• With income	8013
• Average 1995 income $	8014
• Standard error of average income $	8015
• 45-64 years	8016
• Without income	8017
• With income	8018
• Average 1995 income $	8019
• Standard error of average income $	8020
• 65 years and over	8021
• Without income	8022
• With income	8023
• Average 1995 income $	8024
• Standard error of average income $	8025
• Total income of males 15 years and over (20% sample data)	8026
• Without income	8027
• With income	8028
• Average 1995 income $	8029
• Standard error of average income $	8030
• 15-24 years	8031
• Without income	8032
• With income	8033
• Average 1995 income $	8034
• Standard error of average income $	8035
• 25-44 years	8036
• Without income	8037
• With income	8038
• Average 1995 income $	8039
• Standard error of average income $	8040
• 45-64 years	8041
• Without income	8042
• With income	8043
• Average 1995 income $	8044
• Standard error of average income $	8045
• 65 years and over	8046
• Without income	8047
• With income	8048
• Average 1995 income $	8049
• Standard error of average income $	8050
• Total income of females 15 years and over (20% sample data)	8051
• Without income	8052
• With income	8053
• Average 1995 income $	8054
• Standard error of average income $	8055
• 15-24 years	8056
• Without income	8057
• With income	8058
• Average 1995 income $	8059
• Standard error of average income $	8060
• 25-44 years	8061
• Without income	8062

• With income	8063
• Average 1995 income $	8064
• Standard error of average income $	8065
• 45-64 years	8066
• Without income	8067
• With income	8068
• Average 1995 income $	8069
• Standard error of average income $	8070
• 65 years and over	8071
• Without income	8072
• With income	8073
• Average 1995 income $	8074
• Standard error of average income $	8075
• Employment income of population 15 years and over (20% sample data)	8076
• Without employment income	8077
• With employment income	8078
• Average 1995 employment income $	8079
• Standard error of average employment income $	8080
• 15-24 years	8081
• Without employment income	8082
• With employment income	8083
• Average 1995 employment income $	8084
• Standard error of average employment income $	8085
• 25-44 years	8086
• Without employment income	8087
• With employment income	8088
• Average 1995 employment income $	8089
• Standard error of average employment income $	8090
• 45-64 years	8091
• Without employment income	8092
• With employment income	8093
• Average 1995 employment income $	8094
• Standard error of average employment income $	8095
• 65 years and over	8096
• Without employment income	8097
• With employment income	8098
• Average 1995 employment income $	8099
• Standard error of average employment income $	8100
• Employment income of males 15 years and over (20% sample data)	8101
• Without employment income	8102
• With employment income	8103
• Average 1995 employment income $	8104
• Standard error of average employment income $	8105
• 15-24 years	8106
• Without employment income	8107
• With employment income	8108
• Average 1995 employment income $	8109
• Standard error of average employment income $	8110
• 25-44 years	8111
• Without employment income	8112
• With employment income	8113

Average 1995 employment income $	8114
Standard error of average employment income $	8115
45-64 years	8116
Without employment income	8117
With employment income	8118
Average employment income $	8119
Standard error of average employment income $	8120
65 years and over	8121
Without employment income	8122
With employment income	8123
Average 1995 employment income $	8124
Standard error of average employment income $	8125
Employment income of females 15 years and over (20% sample data)	8126
Without employment income	8127
With employment income	8128
Average 1995 employment income $	8129
Standard error of average employment income $	8130
15-24 years	8131
Without employment income	8132
With employment income	8133
Average 1995 employment income $	8134
Standard error of average employment income $	8135
25-44 years	8136
Without employment income	8137
With employment income	8138
Average 1995 employment income $	8139
Standard error of average employment income $	8140
45-64 years	8141
Without employment income	8142
With employment income	8143
Average 1995 employment income $	8144
Standard error of average employment income $	8145
65 years and over	8146
Without employment income	8147
With employment income	8148
Average 1995 employment income $	8149
Standard error of average employment income $	8150
Census family income of all families (20% sample data)	8151
Average 1995 family income $	8152
Standard error of average family income $	8153
Census family income of husband-wife families (20% sample data)	8154
Average 1995 family income $	8155
Standard error of average family income $	8156
Household income of all private households (20% sample data)	8157
Average 1995 household income $	8158
Standard error of average household income $	8159
Household income of one person households (20% sample data)	8160
Average 1995 household income $	8161
Standard error of average household income $	8162
Household income of two or more person households (20% sample data)	8163
Average 1995 household income $	8164

Standard error of average household income $	8165

Table 9 - Family and Household Income, Families (Part II: Social and Economic Data), Occupied Private Dwellings, Household and Housing Costs

Variables	Database Code
Total number of economic families in private households (20% sample data)	9001
Size of economic family, 2 persons	9002
Size of economic family, 3 persons	9003
Size of economic family, 4 persons	9004
Size of economic family, 5 or more persons	9005
Total number of persons in economic families	9006
Average number of persons per economic family	9007
Total number of unattached individuals	9008
Total number census families in private households by number/status of family members in labour force (20% sample)	9009
Number of now-married and common-law couple families	9010
No member in the labour force	9011
Some members in the labour force	9012
One member only	9013
Two or more members	9014
Both spouses/partners in the labour force	9015
Number of lone-parent families	9016
No member in the labour force	9017
Some members in the labour force	9018
Parent in labour force	9019
Total number of occupied private dwellings (20% sample data)	9020
Average number of rooms per dwelling	9021
Average number of bedrooms per dwelling	9022
Average value of dwelling $	9023
Owned	9024
Rented	9025
Band housing	9026
Regular maintenance only	9027
Minor repairs	9028
Major repairs	9029
Period of construction, before 1946	9030
Period of construction, 1946-1960	9031
Period of construction, 1961-1970	9032
Period of construction, 1971-1980	9033
Period of construction, 1981-1990	9034
Period of construction, 1991-1996	9035
Total number of private households by household type (20% sample data)	9036
One-family households	9037
Multiple-family households	9038
Non-family households	9039
Number of persons in private households (20% sample data)	9040
Average number of persons in private households	9041
Tenant one-family households without additional persons	9042

Average gross rent $	9043
Gross rent spending 30% or more of household income on shelter costs	9044
Owner one-family households without additional persons	9045
Average owner's major payments $	9046
Owner's major payments spending 30% or more of household income on shelter costs	9047

Appendix 2

1996 Census of Population - Census Metropolitan Areas (CMA) Spatial Databases

The following section lists the breakdown of each Census Metropolitan Area (CMA) file into their residual components. The “Gea_***” corresponds the Statistics Canada CMA file name for that particular CMA region. Each CMA is broken down into the cities that are contained within that CMA region. The values to the left of the city name correspond to the number of Enumeration Areas for that particular city. The list is divided alphabetically by Province and then alphabetically by City.

Alberta		
❖ Calgary		
	• Calgary	942
	• Gea_825 file	1027
	• Airdrie	16
❖ Edmonton		
	• Edmonton	826
	• Gea_835 file	1254
	• St. Albert	65
	• Fort Saskatchewan	16
	• Leduc	20
	• Spruce Grove	17
❖ Lethbridge		
	• Lethbridge	90
	• Gea_810 file	90
❖ Red Deer		
	• Red Deer	79
	• Gea_830 file	79
British Columbia		
❖ Abbotsford		
	• Abbotsford	150
	• Gea_932 file	207
❖ Kamloops		
	• Kamloops	107
	• Gea_925 file	144
❖ Kelowna		
	• Kelowna	131
	• Gea_915 file	217
❖ Nanaimo		
	• Nanaimo	141
	• Gea_938 file	196
❖ Prince George		
	• Prince George	94
	• Gea_970 file	94
❖ Vancouver		
	• Vancouver	862
	• Gea_933 file	2688
	• Burnaby	269
	• Coquitlam	123

• Langley	34
• New Westminster	94
• North Vancouver	65
• Port Coquitlam	62
• Port Moody	26
• Richmond	217
• Surrey	350
• White Rock	30
❖ Victoria	
• Victoria	169
• Gea_935 file	537
• Colwood	22
Manitoba	
❖ Winnipeg	
• Winnipeg	862
• Gea_602 file	939
New Brunswick	
❖ Moncton	
• Moncton	83
• Gea_305 file	157
❖ Saint John	
• Saint John	123
• Gea_310 file	203
Newfoundland	
❖ St. John's	
• St. John's	156
• Gea_001 file	266
• Mount Pearl	29
Nova Scotia	
❖ Halifax	
• Halifax	206
• Gea_205 file	519
• Dartmouth	96
Ontario	
❖ Barrie	
• Barrie	115
• Gea_568 file	186
❖ Belleville	
• Belleville	59
• Gea_522 file	173
• Trenton	37
❖ Brantford	
• Branford	111
• Gea_543 file	134
❖ Guelph	
• Guelph	139
• Gea_550 file	153
❖ Hamilton	
• Hamilton	441

• Gea_537 file	797
• Burlington	162
• Stoney Creek	63
❖ Kingston	
• Kingston	124
• Gea_521 file	253
❖ Kitchener	
• Kitchener	238
• Gea_541 file	502
• Cambridge	128
• Waterloo	101
❖ London	
• London	502
• Gea_555 file	613
• St. Thomas	43
❖ North Bay	
• North Bay	79
• Gea_575 file	111
❖ Oshawa	
• Oshawa	161
• Gea_532 file	328
❖ Ottawa-Hull	
• Gea_505 file	1403
• Aylmer	40
• Buckingham	16
• Gatineau	118
• Gloucester	120
• Hull	107
• Kanata	54
• Nepean	134
• Ottawa	534
• Vanier	31
❖ Peterborough	
• Peterborough	101
• Gea_529 file	155
❖ Sarnia	
• Sarnia	103
• Gea_562 file	132
• Point Edward	3
❖ Sault Ste. Marie	
• Sault Ste. Marie	110
• Gea_590 file	128
❖ St. Catharines-Niagara	
• Niagara Falls	100
• Gea_539 file	504
• Port Colborne	26
• St. Catharines	180
• Thorold	20

	Welland	61
❖ Sudbury		
	Sudbury	147
	Gea_580 file	243
❖ Thunder Bay		
	Thunder Bay	176
	Gea_595 file	207
❖ Toronto		
	Toronto	1200
	Gea_535 file	5611
	Brampton	277
	Etobicoke	434
	Markham	163
	Mississauga	579
	North York	795
	Oakville	151
	Richmond Hill	112
	Scarborough	705
	Vaughan	136
	York	249
❖ Windsor		
	Windsor	271
	Gea_559 file	378
Quebec		
❖ Chicoutimi-Jonquiere		
	Gea_408 file	219
	Chicoutimi	87
	Jonquiere	74
	La Baie	27
❖ Montreal		
	Gea_462 file	4734
	Anjou	56
	Baie-d'Urfe	4
	Beaconsfield	18
	Beauharnois	11
	Beloeil	20
	Blainville	30
	Boisbriand	26
	Bois-des-Filion	8
	Boucherville	42
	Brossard	70
	Candiac	16
	Carignan	8
	Chambly	23
	Charlemagne	8
	Chateauguay	45
	Cote-Saint-Luc	59
	Delson	8

Dollard-des-Ormeaux	43
Dorval	24
Duex-Montanges	17
Greenfield Park	19
Hampstead	7
Hudson	9
Kirkland	14
La Plaine	17
La Prairie	20
Lachenaie	19
Lachine	54
LaSalle	101
L'Assomption	15
Laval	399
Lavaltrie	7
Le Gardeur	20
Lemoyne	8
Lery	4
L'Ile Bizard	14
L'Ile Cadieux	1
L'Ile Dorval	1
L'Ile Perrot	12
Longueil	180
Lorraine	8
Maple Grove	3
Mascouche	33
Mercier	10
Mirabel	35
Montreal	1735
Montreal-Est	7
Montreal-Nord	113
Montreal-Ouest	5
Mont Royal	22
Mont Saint Hilaire	17
Otterburn Park	8
Outremont	28
Pierrefonds	63
Pincourt	9
Pointe Claire	39
Repentigny	62
Richelieu	5
Rosemere	12
Roxboro	6
Saint Anne de Bellevue	8
Saint Antoine de Lavaltrie	6
Saint Basile le Grand	13
Saint Bruno de Montarville	23
Saint Constant	23

• Sainte Anne des Plaines	17
• Sainte Catherine	17
• Sainte Genevieve	5
• Sainte Julie	27
• Sainte-Marthe-sur-leLac	10
• Sainte Therese	27
• Saint Hubert	89
• Saint-Lambert	31
• Saint-Laurent	115
• Saint-Pierre	9
• Terrabonne	52
• Varennes	21
• Vaudreuil-Dorion	25
• Verdun	114
• Westmount	34
❖ Quebec	
• Quebec	313
• Gea_421 file	1025
• Beauport	94
• Bernieres-Saint-Nicolas	17
• Cap-rouge	16
• Charlesbourg	89
• Charny	13
• Chateau-Richer	7
• Fossambault-sur-le-Lac	2
• Lac-Delage	1
• Lac-Saint-Joseph	1
• L'Anceinne-Lorette	20
• Levis	59
• Lortteville	22
• Sainte-Foy	118
• Saint-Emile	13
• Saint-Jean-Chrysostome	17
• Saint-Redempteur	6
• Saint-Romuald	17
• Sillery	18
• Val-Belair	32
• Vanier	18
❖ Saint-Jean-sur-Richelieu	
• Saint-Jean-sur-Richelieu	79
• Gea_459 file	155
• Iberville	20
• Saint-Luc	30
❖ Sherbrooke	
• Sherbrooke	125
• Gea_433 file	233
• Bromptonville	4
• Fleurimont	20

• Lennoxville	9
• Waterville	4
❖ Trois-Rivieres	
• Trois-Rivieres	88
• Gea_442 file	215
• Becancour	17
• Cap-de-la-Madeleine	43
• Saint-Louis-de-France	11
• Trois-Rivieres-Ouest	30
Saskatchewan	
❖ Regina	
• Regina	237
• Gea_705 file	280
❖ Saskatoon	
• Saskatoon	266
• Gea_725 file	331

Appendix 3

CMA Spatial Databases - Meta Data

An important aspect to all spatial databases are their associated metadata which is the descriptive information explaining what the data in the database are referring to. The following tables explain the metadata coding used by Statistics Canada for the 1996 Census of Population spatial databases used in the current research.

Field Heading	Data description
Area	Area calculation for the polygon and maintained by Arc/Info®. This field is not available in the MapInfo files. The field values are incorrect and need to be recalculated in a GIS package.
Perimeter	Perimeter calculation for the polygon and maintained by Arc/Info®. This field is not available in the MapInfo files. The field values are incorrect and need to be recalculated in a GIS package.
Gea_nnnt	This field is used by Statistics Canada for internal processing.
Gea_nnnt_i	This field is used by Statistics Canada for internal processing.
EAuid	Uniquely identifies an enumeration area (composed of the 2-digit provincial or territorial code, the 3-digit federal electoral district code, and the 3-digit EA code.
SNF	Identifies whether the enumeration area is covered by a street network file. A letter T represents that it is True and a SNF exists and a letter F indicates that a SNF is not available.
PRuid	Identifies the province or territory
CMAuid	Identifies the Census Metropolitan Area or Census Agglomeration

Metadata for CMA Files (Digital Boundary and Cartographic Files) at the EA level.

Field Heading	Data description
Area	Area calculation for the polygon and maintained by Arc/Info®. This field is not available in the MapInfo files. The field values are incorrect and need to be recalculated in a GIS package.
Perimeter	Perimeter calculation for the polygon and maintained by Arc/Info®. This field is not available in the MapInfo files. The field values are incorrect and need to be recalculated in a GIS package.
GCMA_nnnt_	This field is used by Statistics Canada for internal processing.
GCMA_nnnt_i	This field is used by Statistics Canada for internal processing.
water	A value of 1 indicates water and 0 represents islands within the body of water. The water coverage or layer has no names associated to any of the features. Only water features that are large enough to be represented as polygons are present, no line features are available.
Area_m2	This field is an approximate measure of the area of the water features. The field is calculated using Arc/Info® during the creation phase of the database using a Lambert Conic Conformal projection to provide approximate measurement values in meters
perimeter	This field is an approximate measure of the perimeter of the water features. The field is calculated using Arc/Info® during the creation phase of the database using a Lambert Conic Conformal projection to provide approximate measurement values in meters

Metadata for CMA EA water Polygon data files (GCMA_nnnc.ext).

The following table describes the metadata for the EA centroids (representative points) for use with the 1996 Census of Population polygon spatial data sets.

Field Heading	Data description
EAuid	Enumeration Area Unique Identifier. Uniquely identifies an Enumeration Area (composed of the 2-digit provincial or territorial code, the 3-digit federal electoral district code, and the 3-digit EA code. The formula for each EAuid is (PR + FED + EA).
EAPop96	EA 1996 Population. The population counts for a particular area which represents the number of Canadians whose usual residence is in that area.
EAdwell	EA Dwelling Count. Dwelling counts refers to all private dwellings in Canada occupied by their usual residents. The dwelling counts do not include collective dwellings. The population in collective dwellings is included in the population counts.
EA_IR96	Enumeration Area 1996 Census Indian Reserve Refusal Flag. Census enumeration was not permitted on several Indian Reserves thus the Census is incomplete in these areas. Population counts for incomplete enumerated reserves at the EA level have been suppressed and will have a null value.
EAurb_rur	Enumeration Area Urban/Rural Status. The code indicates the urban/rural status of EAs inside and outside Census Metropolitan Areas (CMA)/Census Agglomerations (CA). Code Definition 1 - Urban Core 2 - Urban Fringe 3 - Rural Fringe 4 - Urban Area outside CMAs/CAs 5 - Rural Area outside CMAs/CAs
EALamx	Enumeration Area Lambert Coordinate x axis. Lambert Conformal Conic projection is recommended for maps at small scales.
EALamy	Enumeration Area Lambert Coordinate y axis. Lambert Conformal Conic projection is recommended for maps at small scales.
EALat	Enumeration Area Latitude. Field contains latitude coordinates (in degrees and decimals north of the equator) for each EA representative point.
EALong,	Enumeration Area Latitude. Field contains longitude coordinates (in degrees and decimals west of the prime meridian) for each EA representative point.
PR	Provincial Code. Statistics Canada uses codes to represent provinces and territories. The code is a 2-digit code that uniquely identifies each province/territory and is based on the Standard Geographical Classification (SGC). The code is assigned from east to west. The first digit represents the region in Canada in which the province/territory is located and the second digit denoted one of the ten provinces and two territories. PR Name 10 - Newfoundland 11 - Prince Edward Island 12 - Nova Scotia 13 - New Brunswick 24 - Quebec 35 - Ontario 46 - Manitoba 47 - Saskatchewan 48 - Alberta 59 - British Columbia 60 - Yukon Territory 61 - Northwest Territories

Metadata for all EA Centroids or Representative points files.

FED	Federal Electoral District (FED). FED refers to any place entitled to elect a representative member to serve in the House of Commons. There are 295 FEDs in Canada.
EA	Enumeration Area (EA) code. Each EA is assigned a 3-digit code that is unique within a Federal Electoral District (FED)
ER	Economic Region. An ER is a grouping of complete Census Divisions (CD) with one exception in Ontario.
CD	Census Division. CD is a term applied to areas established by Provincial law which are intermediate geographic areas between the municipality (Census Subdivision) and the Provincial level.
CCS	Census Consolidate Subdivision. CCS is a grouping of Census Subdivisions.
CSD	Census Subdivision. CSD refers to municipalities as determined by provincial legislation.
CMA	Census Metropolitan Areas. A CMA is a grouping of one CMA and adjacent CAs that are socially and economically integrated. An adjacent CMA and CA can be consolidated into a single CMA region if the total commuting interchange between them is equal to or greater than 35% of the employed labour force living in the CA.
PCMA	Primary Census Metropolitan Area. PCMA is a CMA that is a component of a consolidated CMA. PCMA names are typically based on the largest urban center within the PCMA.
CT	Census Tracts. CTs are small geographic units that represent urban or rural neighbourhood-like communities created in al CMAs within an urban core population of 50,000 or more.
UA	Urban Area Code. UAs have a minimum population concentration of 1,000 and a population density of a least 400 per square kilometer. All territories outside UAs are considered rural. UA codes are assigned based on a 4-digit code that is assigned sequentially upon UA creation.
DPL	Designated Place. A Designated Place is an area created by a province to provide services and to structure fiscal arrangements for submunicipal areas which are often unorganized areas. DPL Type — Province LSD Local Service District — Newfoundland, Nova Scotia CFA Class IV Area — Nova Scotia LSB Local Service Board — Ontario NCM Northern Community — Manitoba OHM Organized Hamlet — Saskatchewan UNP Unincorporated Place, — Alberta IST Island Trust — British Columbia
Comment	Field contains address information for selected EAs. The addresses will correspond to a single apartment building, large townhouse complexes, hospital, jail. The data are from the Geographic Attribute Database.

Metadata for all EA Centroids or Representative points files (con't).

Appendix 4

ARCLA Instructions

The ARCLA ArcView extension allows the user to run the *p*-median location-allocation heuristic with data that is exported from the Census Attribute Database and prepared in ArcView. The extension is designed to allow the end user to export data to the location-allocation program and then import the results from the location-allocation program to be displayed in the GIS software. If these instructions are followed carefully, there should be little trouble obtaining nice results. The ARCLA instructions are intended to be used as a brief summary of the procedures to properly use the ARCLA extension. Where the overall intended use of the ARCLA extension is to manipulate and visualize the 1996 Canadian Census of Population data to test location-allocation models. The instructions that follow assume that the user is familiar with ArcView, as a result the notes do not explain many very basic ArcView operations.

Installing the extension

The extension developed for the current research requires the following; ArcView 3.0 or higher installed on Microsoft Windows 95 or higher, minimum of 1 MB of free disk space to install the extension, minimum of 32 MB of RAM. The data sets that are used with the ARCLA extension can be very large and require large amounts of storage space; therefore, it is recommended that the user have 100 MB of free hard disk storage space available. To install ARCLA, copy the file Arcla.avx file to the following directory: \ESRI\AV_GIS30\ArcView\EXT32\. The location-allocation program (longav.exe) needs to be copied to C:\Temp in order for the ARCLA extension to work properly.

Loading the extension

Once the extension is installed in the appropriate location, start ArcView, click on the *File* pull down menu and select the *Extension* menu options. A new dialog box appears, place a check mark to the left of the ARCLA title, this will load the ARCLA extension. Once activated, ARCLA adds several custom menus to the current ArcView project file. By default, the new menus are displayed to the right of the *Help* menu. The titles for the new menus are from left to right as follows: *LA Data Preparation*, *Solving LA Problems*, *Mapping LA Solutions* and *Utilities*. The main ArcView interface is the same, with all the same menus, buttons and tool bars.

Data storage in ARCLA

Data storage is a delicate issue when using ArcView. ArcView is known to save files in obscure locations on the user's hard drive. To counteract ArcView's tendencies, is it strongly recommended by the researcher that a working directory for ArcView be set. A working directory is the directory that ArcView writes all of its underlying files to. By setting a working directory in ArcView, it forces ArcView to use the chosen directory instead some other unknown directory. The default working directory is usually C:\Temp; however, there are exceptions. It is advisable that the ARCLA user use C:\Temp as their working directory and the primary location to store all of their working files. To find ArcView's home directory create a new script and enter in the following information:

```
themessage = "$HOME".AsFileName.GetFullName
theTitle = "ArcView's Current $HOME Directory"
MsgBox.Info(themessage, theTitle)
```

Compile the script by clicking on the check mark button and then execute the script by selecting the run button. A message box appears listing the directory path that is currently set to be ArcView's $HOME directory on that particular computer.

For the data manipulation process to go smoothly using ArcView, the user is strongly encouraged to use C:\Temp to store all of their working files. This directory is used to house all of the files that are created for use with the ARCLA extension. These files include the Census Attribute Database query file and all the spatial database files that are used with the location-allocation heuristic. It is critical that all files are stored in one directory. This method is suggested for several reasons. First, if this method is followed it is very easy to manage all files and the user can be assured that all of the relevant files are stored in one convenient location. Secondly, ArcView does not handle external files in a very efficient manner and by ensuring that all of the working files are stored in one directory the process is be guaranteed to work very efficiently and free of anomalies. It is also suggested that this directory is used for the analysis of one Census variable at a time. This is because the location-allocation program could potentially overwrite the input parameters files, if files for more than one variable is stored in the same directory. It is therefore recommended that the analysis for one variable is finished and the files are removed from the working directory to be backed up before the data files for another variable are copied into the working directory.

Setting Up ARCLA

- Open ArcView, start a new project and load the ARCLA extension (*File - extensions* - select *ARCLA*).

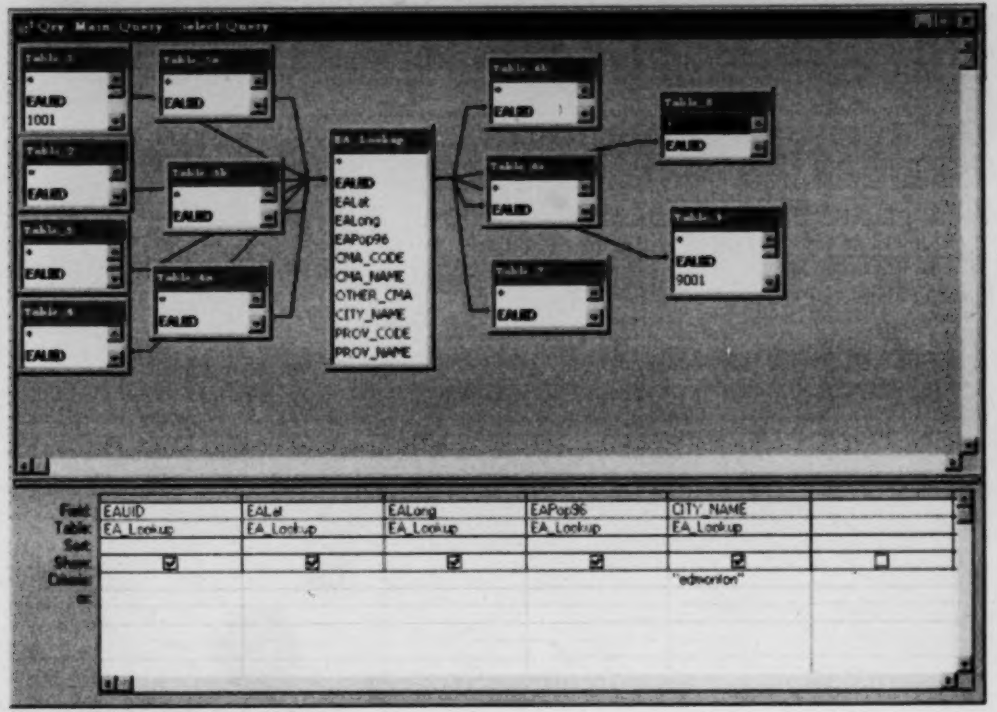
- Create a new view and set your working directory, *file - set working directory*. This step is critical for ARCLA to properly function. It is recommended that C:\Temp always be used as the ARCLA working directory. All files that are to be used in ARCLA are to be stored in this directory.

- The next step requires that data be extracted from the Census Attribute Database for use in ArcView. There are many methods of extracting the appropriate data from the Census Attribute Database. However, only one method is discussed and is recommend for use with the ARCLA extension because of its simplistic nature. This method requires that the user query the Census Attribute Database using the query tools in the database and then export the query results in a database file format (*.dbf).

- The database query requires that the following information from the Census Attribute database is included: the EAuid field, EALat field, EALong field, the 1996 Total Population field (EAPop96), City_Name field (query for the city of interest), and a Census demographic variable. A Census demographic variable is required and is used as the demand weight in the location-allocation program. The demand weight variable is derived from one of the demographic variables in one of the Nation tables stored in the Census Attribute Database. All of the other required fields come from the EA_Lookup Table.

- The user is able to quickly query the Census Attribute Database because a query is already setup in the Census Attribute Database called qry_Main_Query. All the user needs to do is select a Census demographic variable that is to be used in the location analysis by dragging and dropping the field(s) from the appropriate Nation Tables down into the query design grid (beside the City_Name box). Also, the user needs to type in the name of the city they would like to use in the criteria area under the City_Name box. The figure below show the qry_Main_Query window and shows Edmonton typed in as the requested city.

- Once the query is run, it should be exported in a database format (dbf) file to the working directory. Once the query is saved in the proper format, the user is now ready to initiate ArcView and begin the data manipulation procedures. This query is referred to as the Census query table hereafter.


Qry Main Query : Select Query
Table_1
EAUID
1001
Table_2
Table_3
Table_4
Table_5a
Table_5b
Table_6a
EA_Lookup
EAUID
EALat
EALong
EAPop96
CMA_CODE
CMA_NAME
OTHER_CMA
CITY_NAME
PROV_CODE
PROV_NAME
Table_6b
Table_6c
Table_7
Table_8
Table_9
9001

Field:	EAUID	EALat	EALong	EAPop96	CITY_NAME	
Table:	EA_Lookup	EA_Lookup	EA_Lookup	EA_Lookup	EA_Lookup	
Sort:						
Show:	☑	☑	☑	☑	☑	☐
Criteria:					"edmonton"	
or:						

Microsoft Access query window.

Importing the Census query table into ArcView

- Add the Census query table to the ArcView project.

- Once the Census query table is available in ArcView, the table needs to be converted from a database file (dbf) into an ArcView Event Theme. A tabular database file can be converted to a point spatial data file through the use of an ArcView function called an Event Theme. The new point spatial data file is referred to as an event theme and is constructed from a table that contains geographic locations such as latitude, longitude or x, y. When an event theme is created, the geographic locations from the tabular data table are converted into an ArcView point spatial data format. ARCLA provides a function called *Convert Census query table to Event Theme* function under the *LA data preparation*. The function will ask the user to *Choose a Table*, the user should select the Census query table. After the program is finished there will be a new point theme in the current view that is based on the coordinates stored in the Census query table.

Create the required Location-Allocation files

- The location-allocation program requires four input files. Two of these files are created from the Census query table's point theme (presently located in the project's view window). The other two files are the programs run parameters files.

- Of the two files that are made from the Census query table's point theme, one is used as the demand locations, the other file is the potential facility locations. The user must first determine which EA zones to use as their demand points. To do this, the chosen EA zones must first be highlighted in ArcView (typically yellow in colour). The reason for this is that export demand function will only export the selected EA zones. To make the demand file use the function under the *Data Preparation* menu called *Create Demand file for LA program.* The program asks the user to select a theme to be used as their demand nodes, the Census query table's point theme should be selected. The program then asks which field is used as the demand weight, select the appropriate Census variable field. The result is that the program will create a text file with all of the data from the attribute table of the Census query table point theme. The demand file requires the following fields: EAuid, latitude (EALat), longitude (EALong) and a demand variable. The file will be called [variable name]_demand_points.txt

- Next a potential facility location file needs to be created. The user must first determine which EA zones to use as their facility points. To do this, the chosen EA zones must first be highlighted in ArcView (typically yellow in colour). The reason for this is that export facility function will only export the selected EA zones. The user is able to export all of the centroids for use as the potential facilities or the user can create some type of query to narrow down the number of candidate locations if they wish. Use the *Create Facility File for LA Program* function located under the *Data Preparation* menu to export the following fields: EAuid, latitude (EALat), longitude (EALong). Check to ensure it saved the text file in your working directory. The file will be called [variable name]_potential_sites.txt

- The remaining two files need to instruct the location-allocation program on how to run (setup parameters). The first file stores the run time parameters used by the location-allocation program. To create the parameters file use the C*reate Parameters File function under the Solving LA Problems* menu. For the number of facilities the user is required to type in a range of facility numbers. For example, a range typed in as 3 5 will result in solutions for 3, 4, and 5 facilities being calculated by the location program and 5 5 will result in a solution only 5 facilities. The location-allocation parameters text file requires the above information to be input in the following format: number of random starts, lowest value for the number of facilities and the highest value for the number of facilities with a single space being placed between each input parameter. The dialog window also requires the user to enter in a Census variable title, the name should correspond to the variable used as the demand weight.

- At this point, there are four text files located in the working directory. They are called: [variable name]_demand_points.txt, [variable name]_potetnial_sites.txt, optimize_name.txt and [variable name]_input_parameters.txt. Three of these files require that they have the same name where it states [variable name].

- To check that the three files have the proper [variable name] use the ARLCA function *Check LA Setup* under the *Solving LA Problems* menu. Enter in the Census variable name when prompted. If it states every thing is okay then continue. Otherwise, ensure you have four files in your working directory and three of the text files all begin with the [variable name] name and the fourth file should be called optimize_name.txt.

- To run the location-allocation program (Longav.exe) from within ArcView use the *Solve Teitz-Bart Heuristic* under the *Solving LA Problems* menu. This will only work if the longav.exe is located it the C:\Temp directory.

- After the location-allocation program has finished its computations, the solution files are ready to be imported ArcView.

Importing the Location-Allocation results into ArcView

- Use the *Mapping LA Solutions – Import Facilities File* to import the proper facility solution file. Select the [variable name]_facility_location.txt file when prompted. Select *yes* when asked to join to an existing shapefile – select the Census query table's point theme as the theme to join the solution file to and follow the remaining instructions.

- Use the *Mapping LA Solutions – Import Demand File* to import the proper facility solution file. Select the [variable name]_demand_allocations.txt file when prompted. Select *yes* when ask to join to an existing shapefile – select Census query table's point theme as the theme to join the solution file to and follow the remaining instructions.

- Next, the solution shapefiles need to be queried to extract only the information relevant to each solution. For example from both the facility_location and demand_allocations solution files we need to extract only those records that correspond to the p=1 (or from facility 1, 2 and so on). For example the value of p=1 in the facility file corresponds to the p=1 in the demand file. Use the respective programs (*Query Facilities Solution File* or *Query Demand Solution File*) found under the *Mapping LA Solutions* menu to query the files for certain values of p, such as 5. It is recommended that useful files names be used to the resulting queried files. For example, call the facility file for 5 facilities - 5_facility and the associated demand file - 5_demand.

- The last step, involves the process of drawing lines between each facility and its respective demand nodes. Use the program *Create Desire Lines between Facilities and Demands* under the *Mapping LA Solutions* menu. Select the facility for one solution, such as 5_facility and the corresponding demand file, 5_demand. Make sure that these two themes do not contain any joins or that these files are not text files but rather they must be ArcView shapefiles.

ARCLA Choropleth Mapping

- A powerful component of a GIS for location-allocation is its ability to create choropleth maps. To do so in ARCLA, add the EA Census spatial database file for the city of interest and water boundaries to the view. Join the Census query table (source) to the City EA attribute table (destination) using the EAuid field in both tables as the common linking field. The data are now joined together for thematic mapping purposes. It is recommended that the variable used as the demand weight is normalized by the total population (EAPop96). To create a choropleth map use the ARCLA function *Create a Choropleth Map for Polygon Themes* under the *Mapping LA Solutions* menu. The program will automatically set the legend to a graduate colour type with a quantile distribution using the number of classes established by the user with the classification variable normalized by the second variable selected.

- Display the overall solutions with desire lines, demand nodes and facilities visible. These three layers should be displayed over top of the corresponding variable choropleth map.

ARCLA Extra Functions

- **Add Latitude/Longitude to user created point theme** A program is included in the ARCLA extension to calculate the latitude and longitude coordinates for a user created point theme. The program is located under the *Data Preparation* menu called *Add Lat, Long Coordinates to Point Theme*. This function allows the user the ability to create their own demand or facility nodes and determine the latitude and longitude coordinates for each node so the these data can be used with the external location-allocation program which reads latitude/longitude coordinates. In order for this program to work correctly, it requires that the active theme(s) are stored in an unprojected view and that a Census spatial database file is present in the view. To change the projection properties of a view to unprojected, select the *Properties* option under the *View* menu and select the *Projection* button. Ensure that the Projection Properties are set to Standard, the Variable field is set to Projections of the World and Type is set to Geographic.

- **Changing Legends** - Use the programs under *Mapping LA Solutions* to change the legends of these two new files (*Change Legend for Facility Theme* or *Change Legend for Demand Themes*). The facility programs give each facility node a unique colour and symbol and demand theme program allows the user to modify the legend of the demand nodes theme into a graduated point symbol map using quantile classes.

- **Display on Selected Fields** - This function is included in ARCLA to allow the user to manually turn on or off (hide) fields quickly. The utility is called *Display only selected Fields in a Table* and is located under the *Utilities* menu.

- **Find Duplicate values** - A useful tool is the *Find Duplicate Values* program housed under the *Utilities* menu. For example, the utility could be used to find duplicate EAuid values. It has been previously pointed out that this is extremely important when working with cities located near larger bodies of water. The program identifies any duplicate values in the selected field by identifying them in a new field called match. If the values in the match field are listed as false, then the value does not have a duplicate in the selected column. Conversely, a record containing the word true indicates that the value in that particular row has a duplicate in the same field somewhere else in the table.

- **Map projections** - Because the world isn't flat but the computer screen is, ARCLA contains tools to accurately display the spatial data in the most meaningful way possible. Map projections attempt to preserve the properties of area, shape, distance and direction while trying to minimize the amount of distortion throughout the spatial database. In order to properly project the 1996 Census Canada spatial data, the data needs to be projected using the Lambert Conformal Conic projection using a spheroid of GRS 80. In order to calculate proper distances in a GIS, the map units need to correspond to the spatial data sets and the proper distance units for the 1996 Census Canada spatial data are in meters. Use the ARCLA function *Set View's Projection* under the *Utilities* menu.

- **Random Selection in a Theme** - Program is found under the *Utilities* menu and it will prompt the user to enter in a percentage and the program will then randomly select that percentage from the active theme.

- **Return Legend to Default Settings** - Use the program under *Mapping LA Solutions* menu to return the legend of any point, line or polygon theme to its default settings.

- **Turn On/Off Multiple Themes** - Use this function to quickly turn on or off any themes in the table of contents, this feature is very useful when dealing with many themes in a view. Program is found under the *Utilities* menu.

ARCLA Notes

- **Area fields** - A cautionary note when using the area field for normalization purposes, the values are incorrect and must be recalculated in ArcView. Use the ARCLA function that is called *Calculate Area for a Polygon Theme* and is located under the *Mapping LA Solutions* menu.

- **Census query table fields** - In order for the ARCLA extension to work properly, the Census query table requires that the fields listed above and in the database file that is imported into ArcView. The reason is that many of the ARCLA functions read from these fields and the functions will not work properly if these fields are missing.

- **Default File names** - It is essential that the input files use the proper naming scheme described above. The location-allocation program will not work properly if the proper name are not used. Also ensure that there are no spaces in any of the file names.

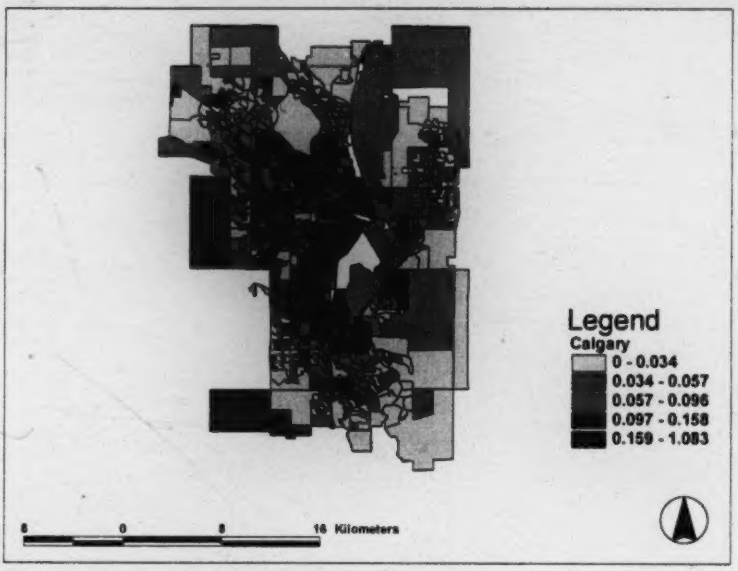
- **Dealing with Queries** - It is strongly recommended that any queried selections be saved as a new shapefile. As well, all text and joined tables should be saved as new shapefiles to preserve the data structure effectively.

- **EAuid field** - Several factors need to be mentioned that might affect the outcome of the ARLCA extension if they are not taken into account. The EAuid field is exported from ArcView in the two input text files as string formatting for use with the location-allocation program. Unfortunately, the EAuid field is returned as a number field in the final solution files from the location-allocation program. This complicates the procedure when the user would like to join the solution data files to an existing Census query table's point theme. However, this problem can be overcome in ArcView by creating a new field in the Census query table's point theme and then converting the EAuid field from string to numerical formatting in the new field using the calculate function under the field menu. The field conversion step is required because all databases require a common data field to be joined together. The common fields must be in the same type of formatting (whether it is text or numerical, both fields need to be in the same formatting) consequently, EAuid must be converted to numerical values. An ARCLA program performs this task for the user and is called Convert EAuid Field to Number and is located under the *Mapping LA Solutions* menu.

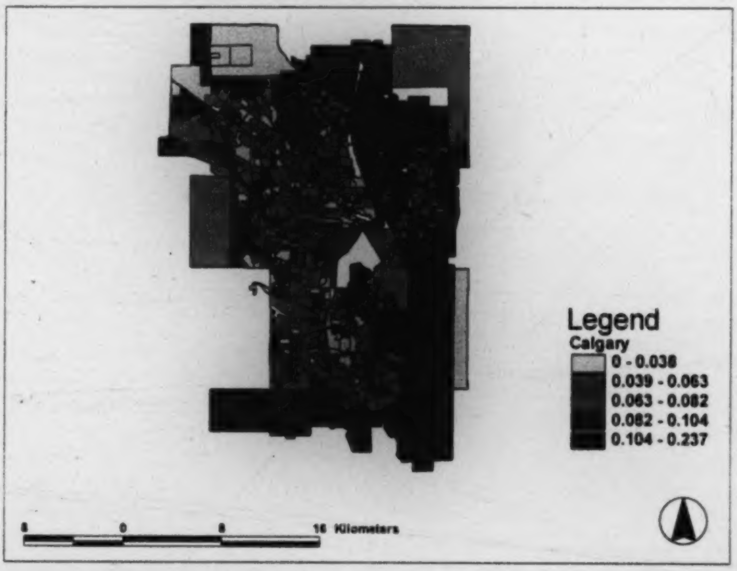
- **Tabular text files** - ArcView, or in particular Avenue, has difficulty dealing with tabular text files as a data source, and because ARCLA is Avenue driven it is highly recommended that any joins performed between a text file and a shapefile be converted to a new shapefile. This technique does not adhere to proper database management principles; however, it ensures that the ARCLA programs can easily read the data sets and efficiently process the tables to obtain nice results.

Appendix 5

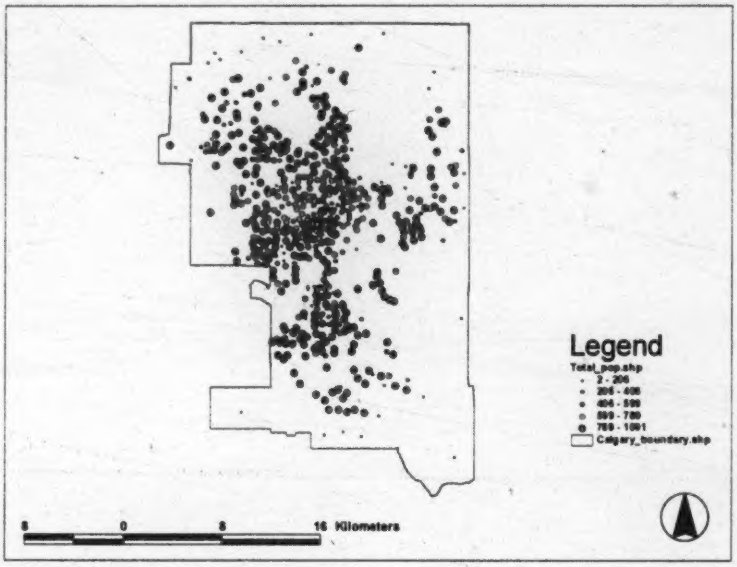
Location-Allocation Analysis - Per Capita Distribution Maps for the 15 selected cities


Legend
Calgary
0 - 0.034
0.034 - 0.057
0.057 - 0.096
0.097 - 0.158
0.159 - 1.083
8
0
8
16 Kilometers

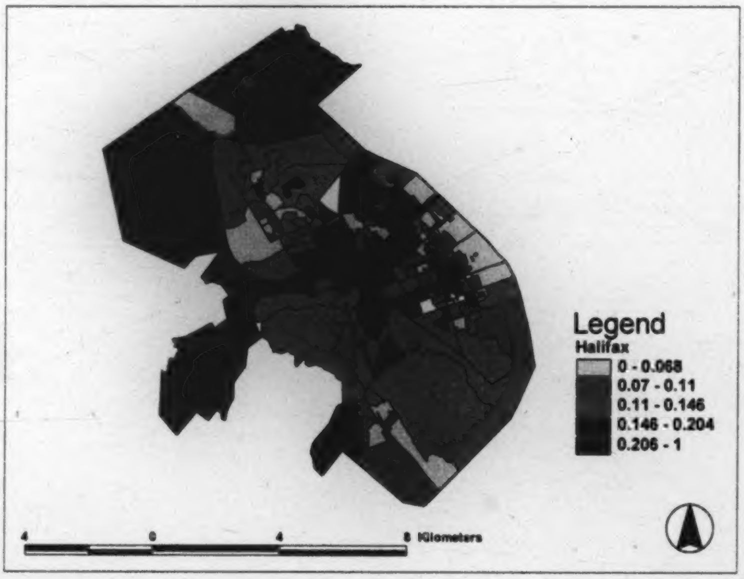
Senior's population per capita in Calgary.


Legend
Calgary
0 - 0.038
0.039 - 0.063
0.063 - 0.082
0.082 - 0.104
0.104 - 0.237
8
0
8
16 Kilometers

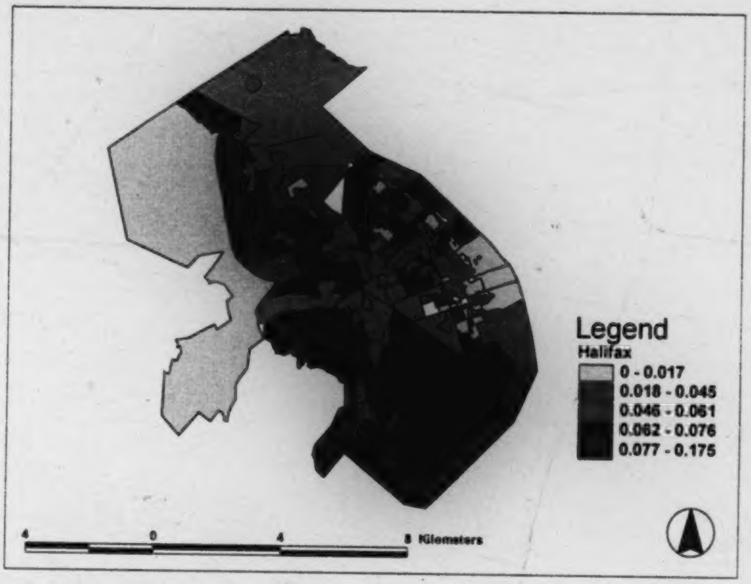
Children's population per capita in Calgary.


Legend
Total_pop.shp
2 - 206
206 - 406
406 - 599
599 - 780
780 - 1001
Calgary_boundary.shp
8
0
8
16 Kilometers

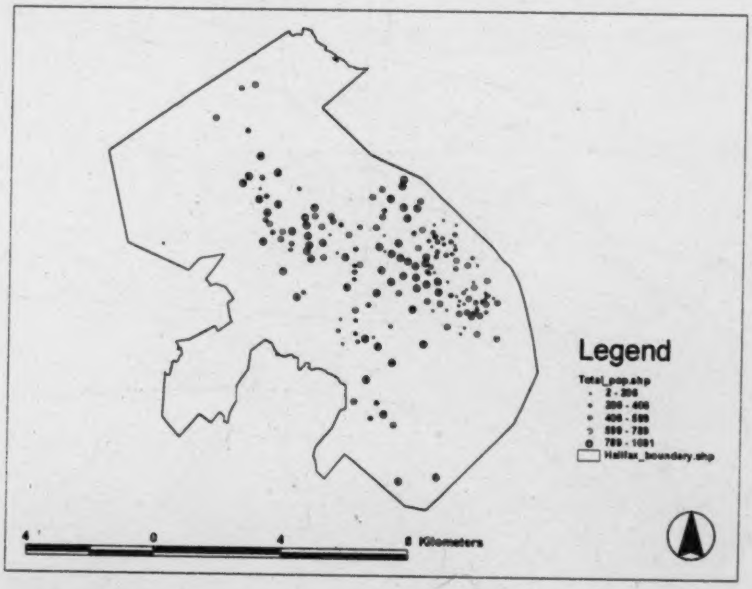
Total population in Calgary.


Legend
Halifax
0 - 0.068
0.07 - 0.11
0.11 - 0.146
0.146 - 0.204
0.206 - 1
4
0
4
8 Kilometers

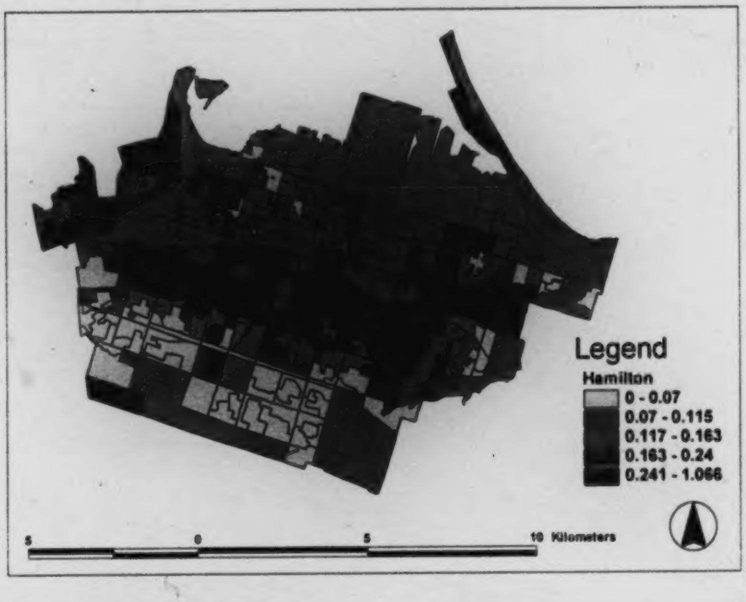
Senior's population per capita in Halifax.


Legend
Halifax
0 - 0.017
0.018 - 0.045
0.046 - 0.061
0.062 - 0.076
0.077 - 0.175
4
0
4
8 Kilometers

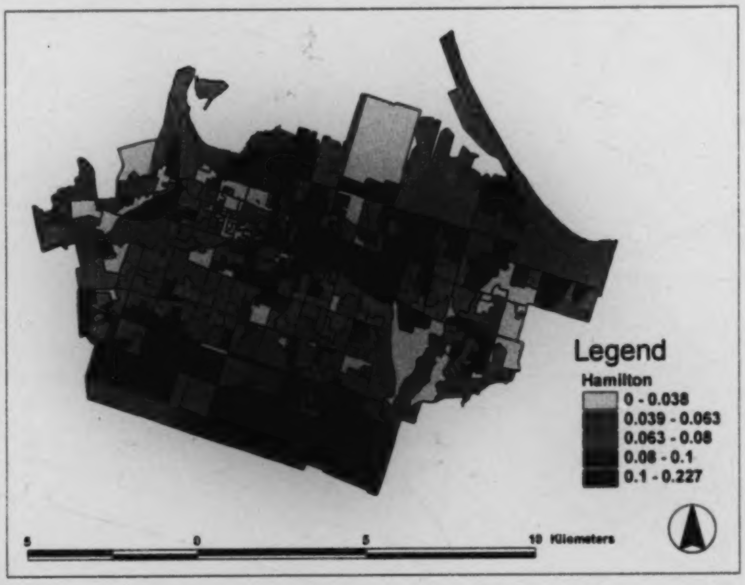
Children's population per capita in Halifax.


Legend
Total_pop.shp
789 - 1081
Halifax_boundary.shp
4
0
4
8 Kilometers

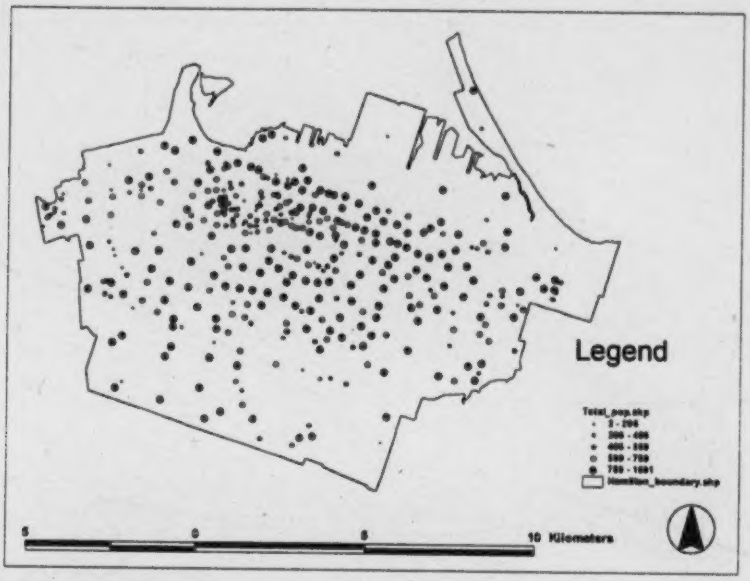
Total population in Halifax.


Legend
Hamilton
0 - 0.07
0.07 - 0.115
0.117 - 0.163
0.163 - 0.24
0.241 - 1.066
5
0
5
10 Kilometers

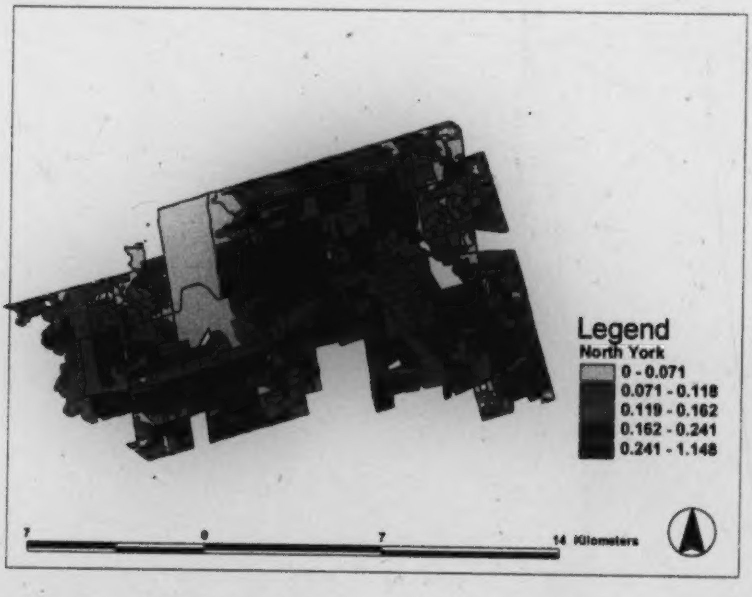
Senior's population per capita in Hamilton.


Legend
Hamilton
0 - 0.038
0.039 - 0.063
0.063 - 0.08
0.08 - 0.1
0.1 - 0.227
5
0
5
10 Kilometers

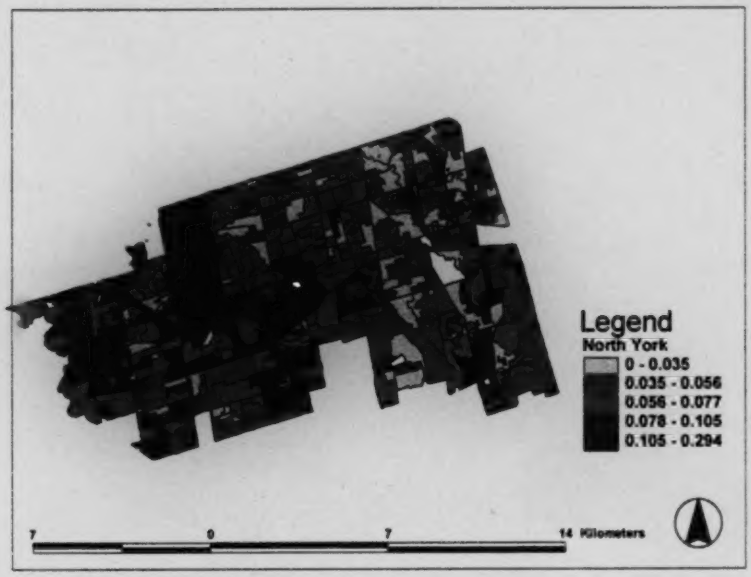
Children's population per capita in Hamilton.


Legend
Total_pop.shp
Hamilton_boundary.shp
5
0
5
10 Kilometers

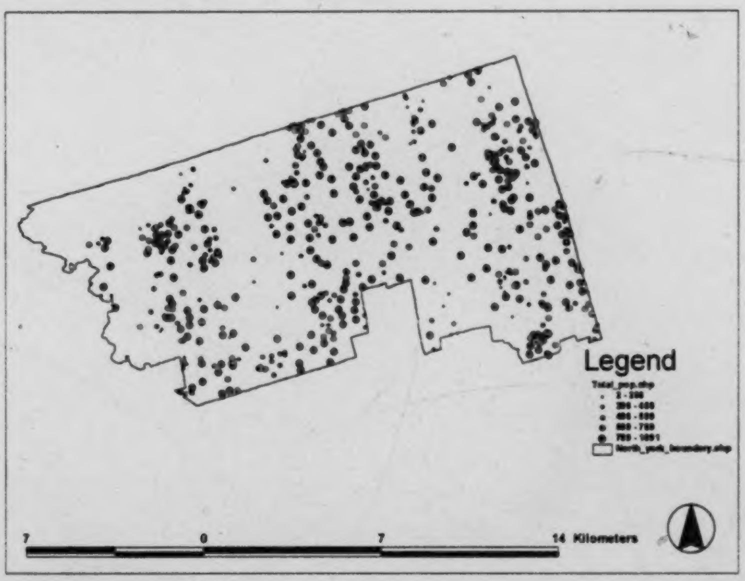
Total population in Hamilton.


Legend
North York
0 - 0.071
0.071 - 0.118
0.119 - 0.162
0.162 - 0.241
0.241 - 1.148
7
0
7
14 Kilometers

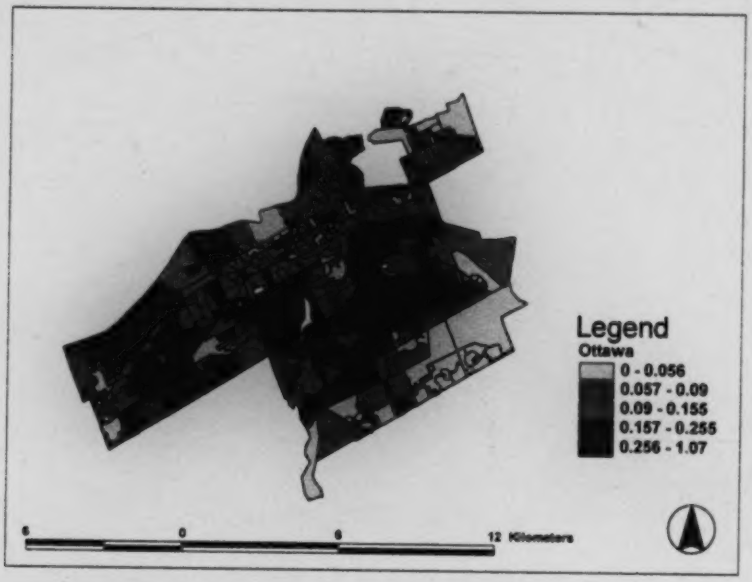
Senior's population per capita in North York.


Legend
North York
0 - 0.035
0.035 - 0.056
0.056 - 0.077
0.078 - 0.105
0.105 - 0.294
7
0
7
14 Kilometers

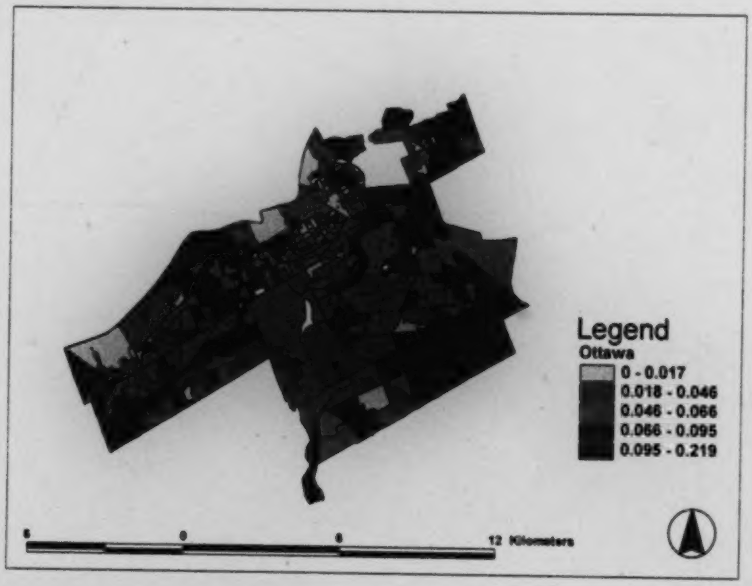
Children's population per capita in North York.


Legend
Total_pop.shp
North_york_boundary.shp
7
0
7
14 Kilometers

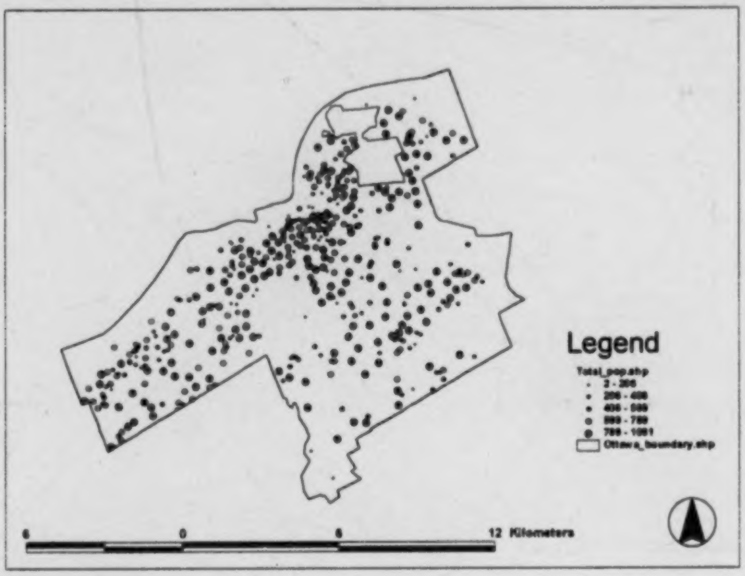
Total population in North York.


Legend
Ottawa
0 - 0.056
0.057 - 0.09
0.09 - 0.155
0.157 - 0.255
0.256 - 1.07
6
0
6
12 Kilometers

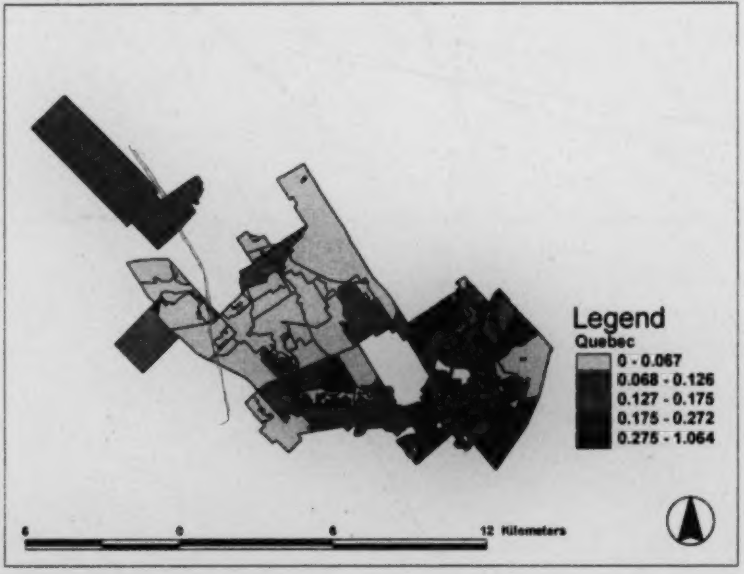
Senior's population per capita in Ottawa.


Legend
Ottawa
0 - 0.017
0.018 - 0.046
0.046 - 0.066
0.066 - 0.095
0.095 - 0.219
6
0
6
12 Kilometers

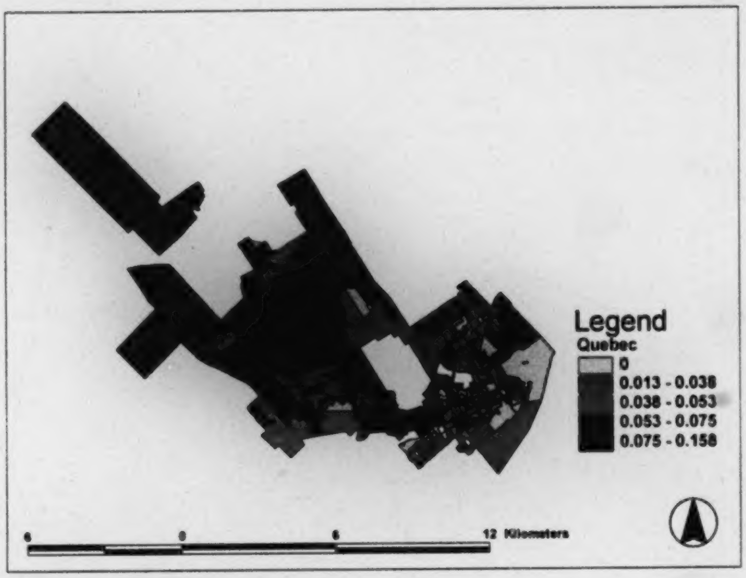
Children's population per capita in Ottawa.


Legend
Total_pop.shp
Ottawa_boundary.shp
6
0
6
12 Kilometers

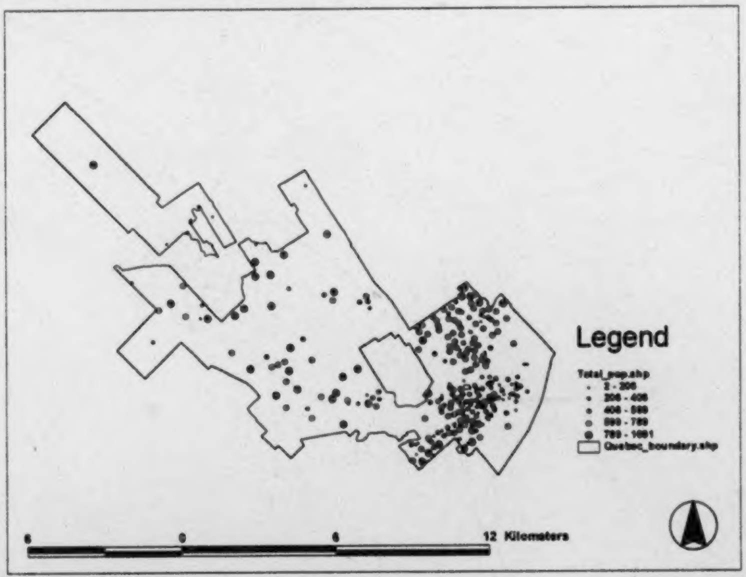
Total population in Ottawa.


Legend
Quebec
0 - 0.067
0.068 - 0.126
0.127 - 0.175
0.175 - 0.272
0.275 - 1.064
6
0
6
12 Kilometers

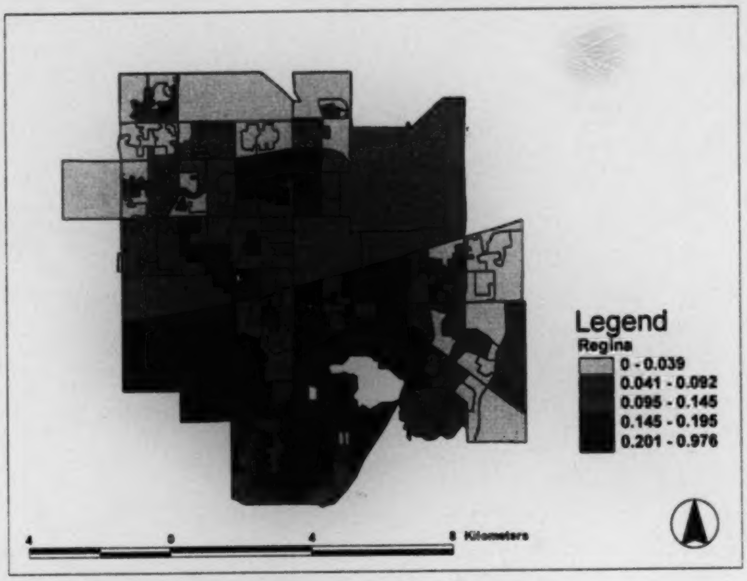
Senior's population per capita in Quebec.


Legend
Quebec
0
0.013 - 0.038
0.038 - 0.053
0.053 - 0.075
0.075 - 0.158
6
0
6
12 Kilometers

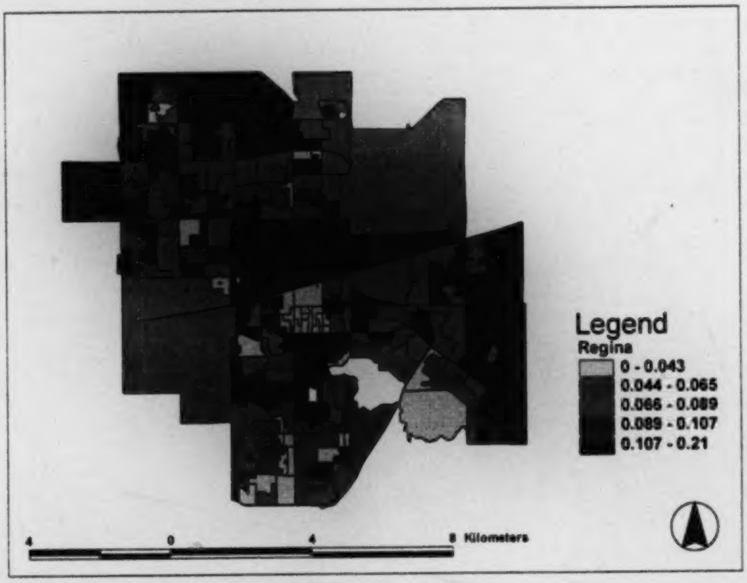
Children's population per capita in Quebec.


Legend
Total_pop.shp
Quebec_boundary.shp
6
0
6
12 Kilometers

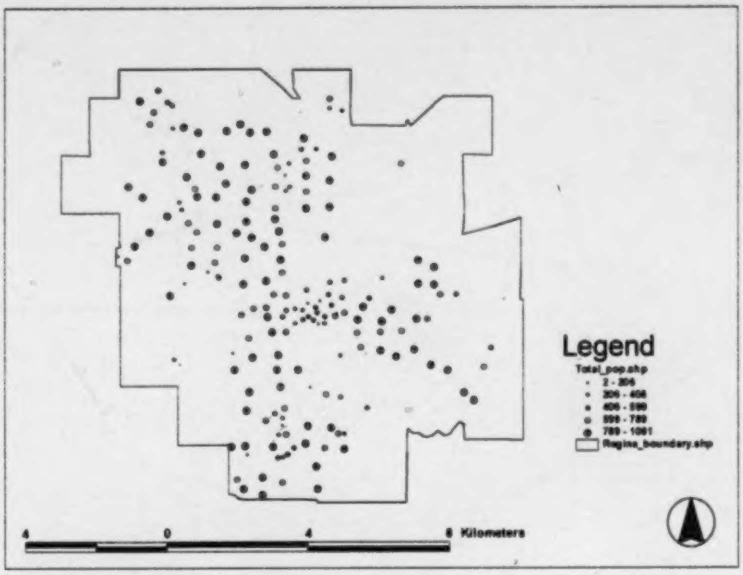
Total population in Quebec.


Legend
Regina
0 - 0.039
0.041 - 0.092
0.095 - 0.145
0.145 - 0.195
0.201 - 0.976
4
0
4
8 Kilometers

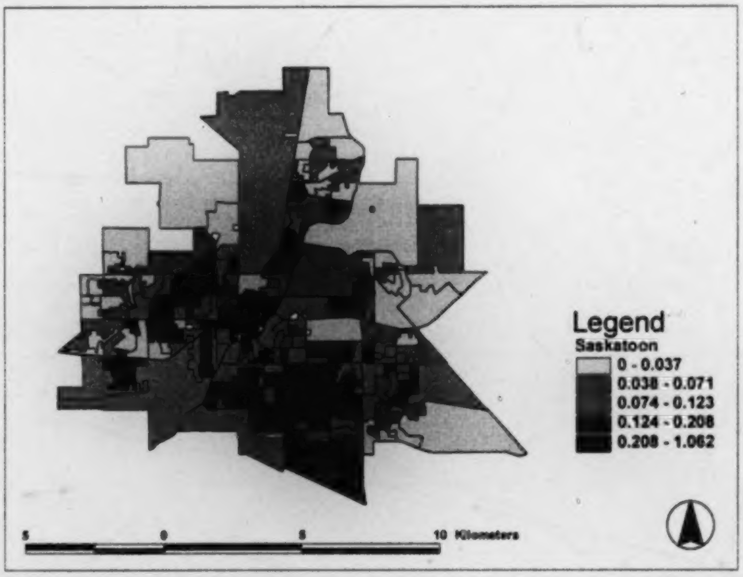
Senior's population per capita in Regina.


Legend
Regina
0 - 0.043
0.044 - 0.065
0.066 - 0.089
0.089 - 0.107
0.107 - 0.21
4
0
4
8 Kilometers

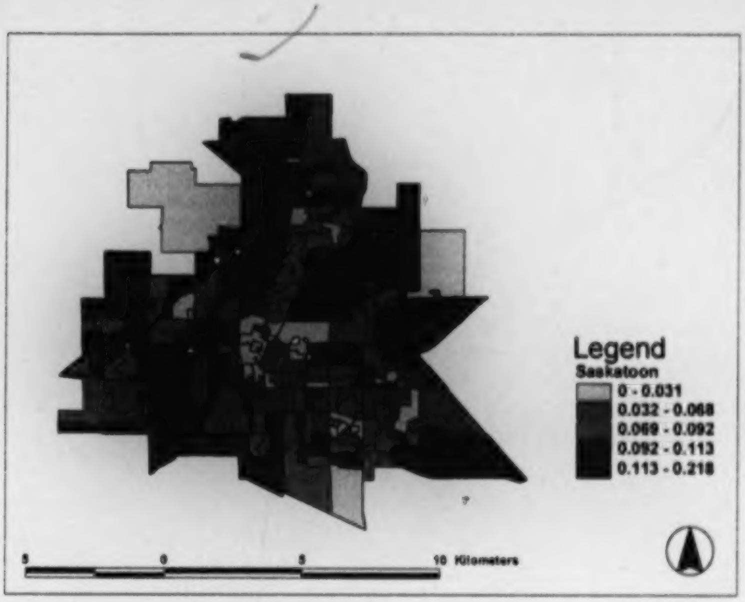
Children's population per capita in Regina.


Legend
Total_pop.shp
2 - 204
205 - 405
406 - 590
591 - 789
789 - 1061
Regina_boundary.shp
4
0
4
8 Kilometers

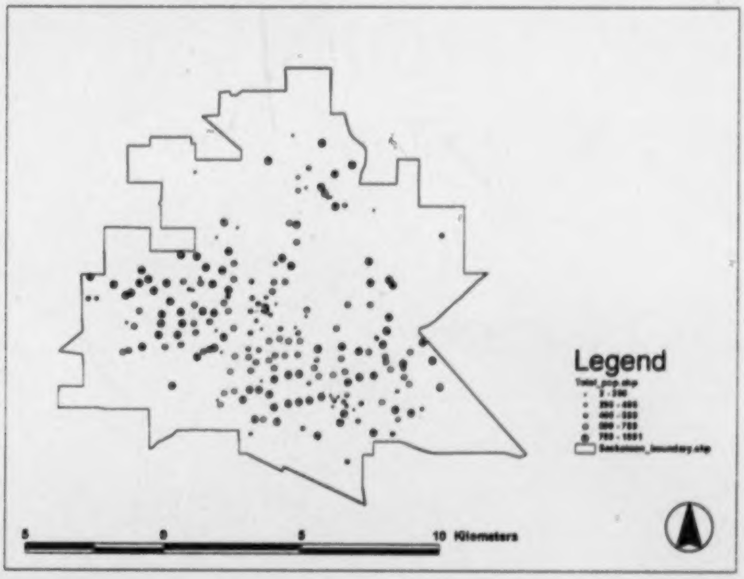
Total population in Regina.


Legend
Saskatoon
0 - 0.037
0.038 - 0.071
0.074 - 0.123
0.124 - 0.208
0.208 - 1.062
5
0
5
10 Kilometers

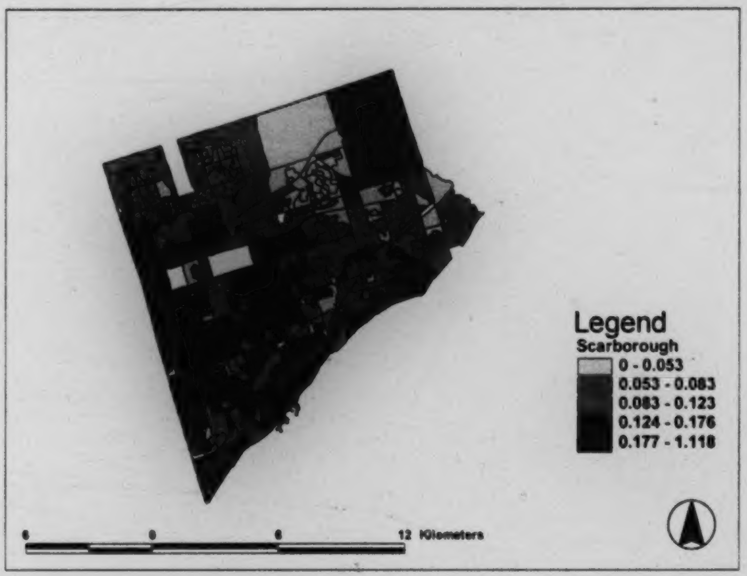
Senior's population per capita in Saskatoon.


Legend
Saskatoon
0 - 0.031
0.032 - 0.068
0.069 - 0.092
0.092 - 0.113
0.113 - 0.218
5
0
5
10 Kilometers

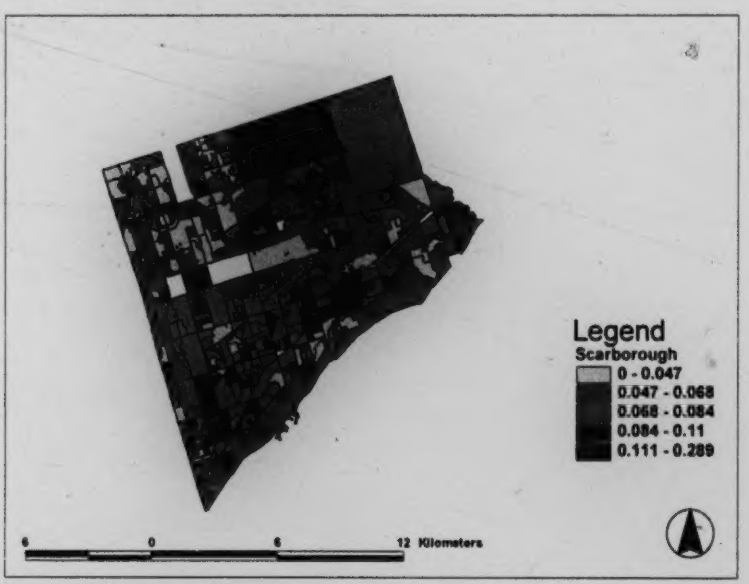
Children's population per capita in Saskatoon.


Legend
Total_pop.shp
Saskatoon_boundary.shp
5
0
5
10 Kilometers

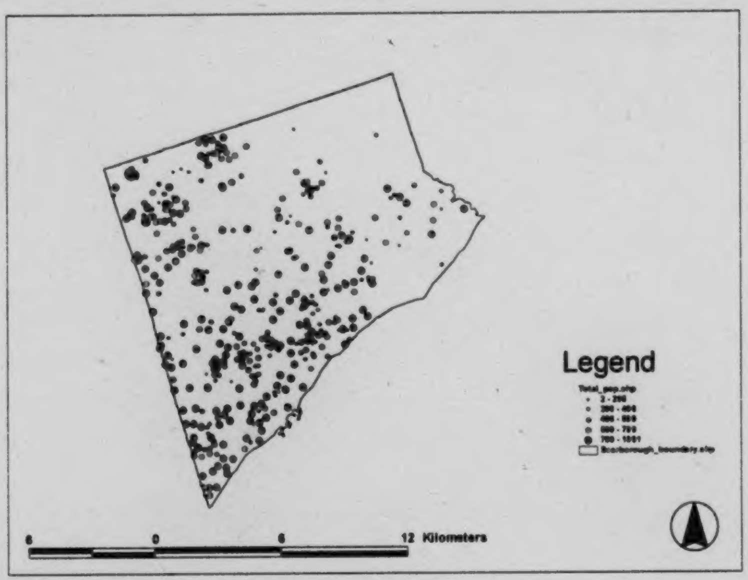
Total population in Saskatoon.


Legend
Scarborough
0 - 0.053
0.053 - 0.083
0.083 - 0.123
0.124 - 0.176
0.177 - 1.118
6
0
6
12 Kilometers

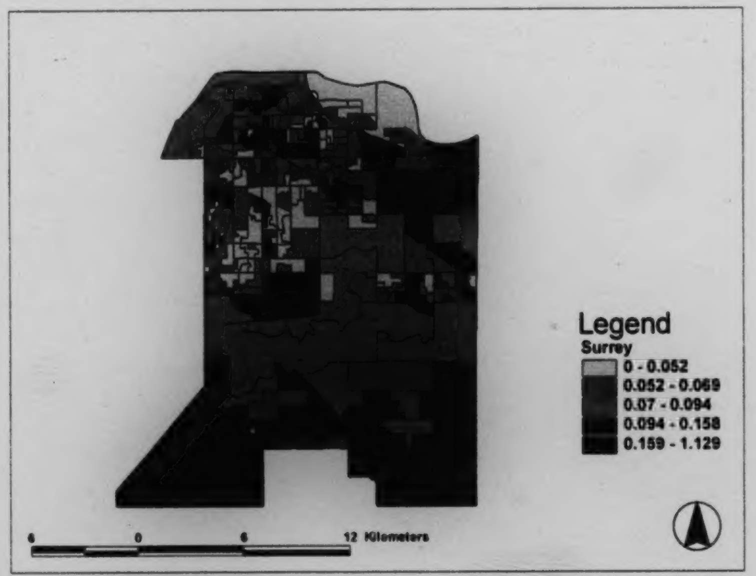
Senior's population per capita in Scarborough.


Legend
Scarborough
0 - 0.047
0.047 - 0.068
0.068 - 0.084
0.084 - 0.11
0.111 - 0.289
6
0
6
12 Kilometers

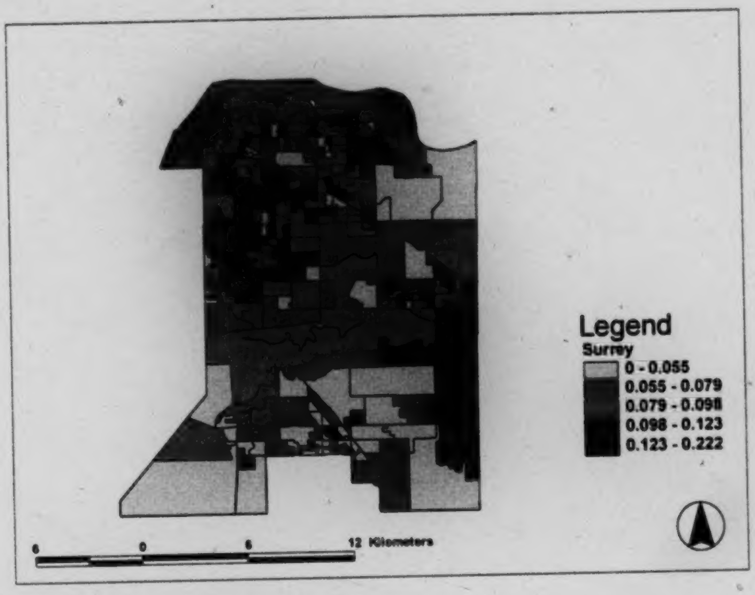
Children's population per capita in Scarborough.


Legend
Total_pop.shp
Scarborough_boundary.shp
6
0
6
12 Kilometers

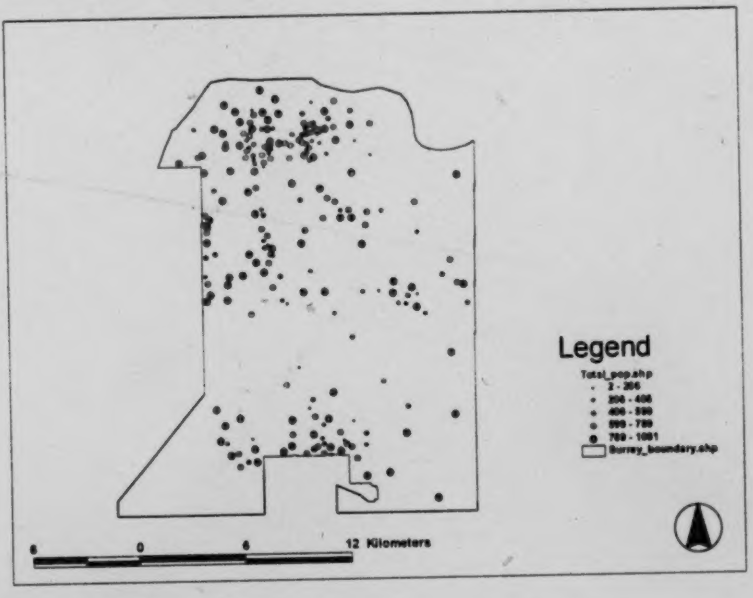
Total population in Scarborough.


Legend
Surrey
0 - 0.052
0.052 - 0.069
0.07 - 0.094
0.094 - 0.158
0.159 - 1.129
6
0
6
12 Kilometers

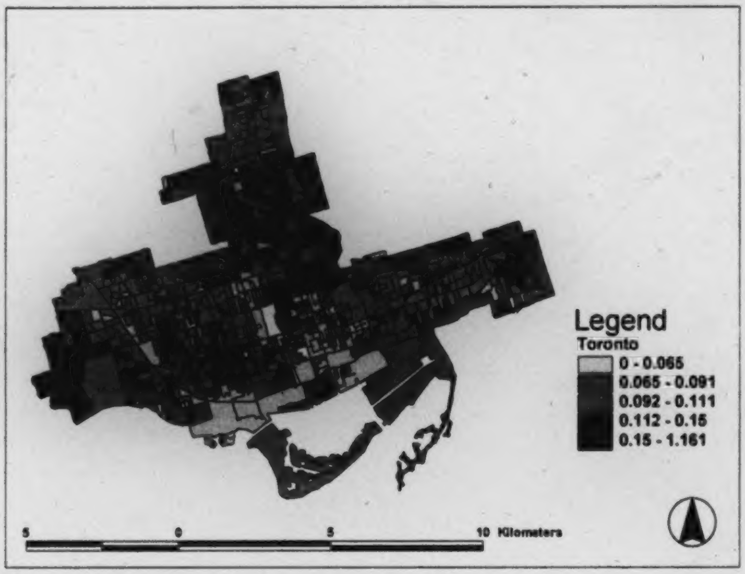
Senior's population per capita in Surrey.


Legend
Surrey
0 - 0.055
0.055 - 0.079
0.079 - 0.098
0.098 - 0.123
0.123 - 0.222
6
0
6
12 Kilometers

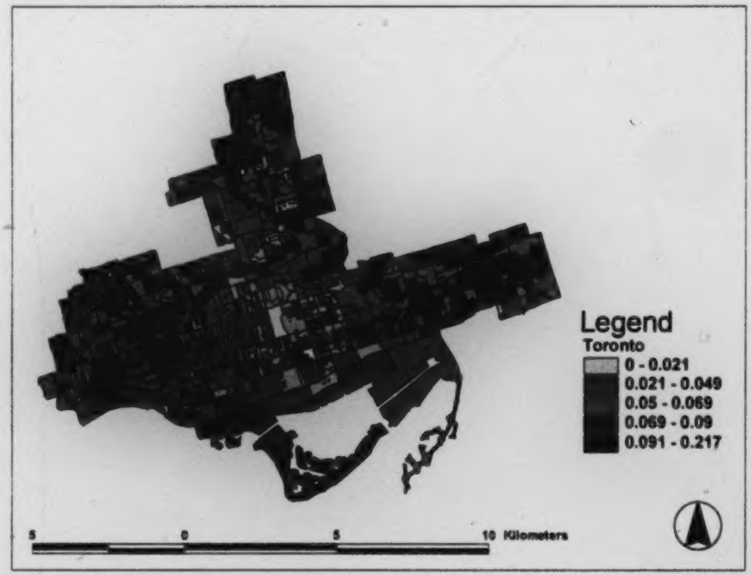
Children's population per capita in Surrey.


Legend
Total_pop.shp
Surrey_boundary.shp
6
0
6
12 Kilometers

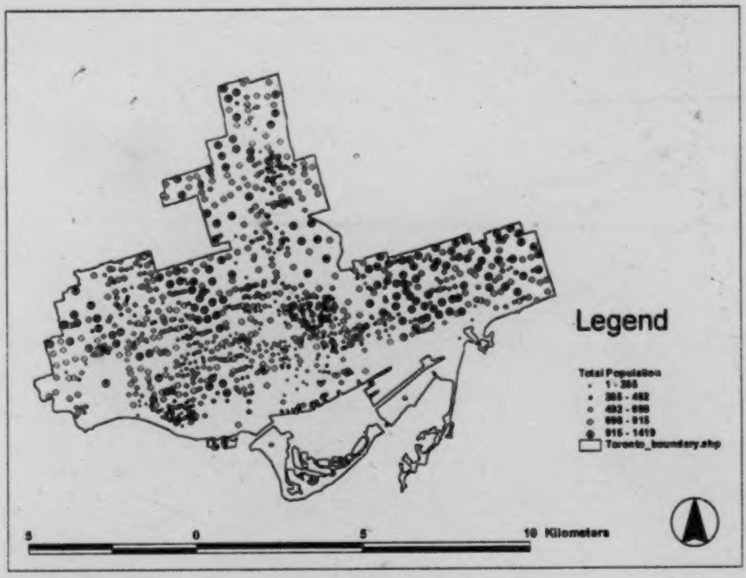
Total population in Surrey.


Legend
Toronto
0 - 0.065
0.065 - 0.091
0.092 - 0.111
0.112 - 0.15
0.15 - 1.161
5
0
5
10 Kilometers

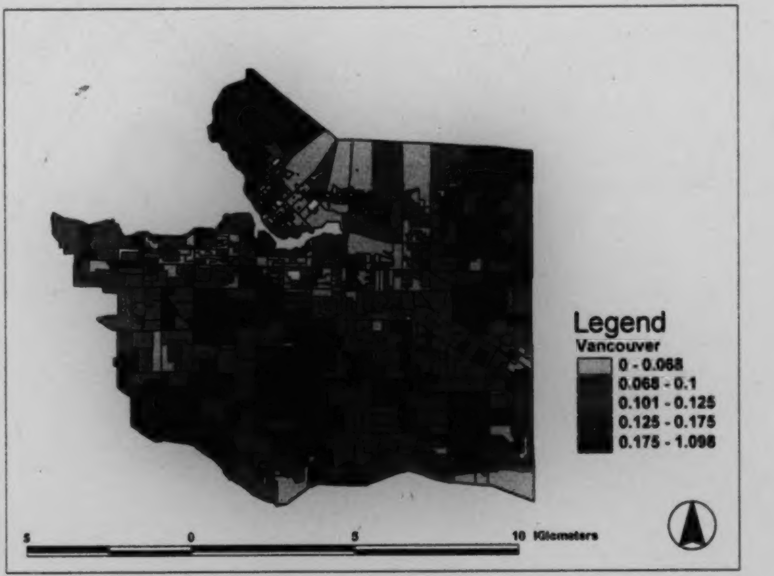
Senior's population per capita in Toronto.


Legend
Toronto
0 - 0.021
0.021 - 0.049
0.05 - 0.069
0.069 - 0.09
0.091 - 0.217
5
0
5
10 Kilometers

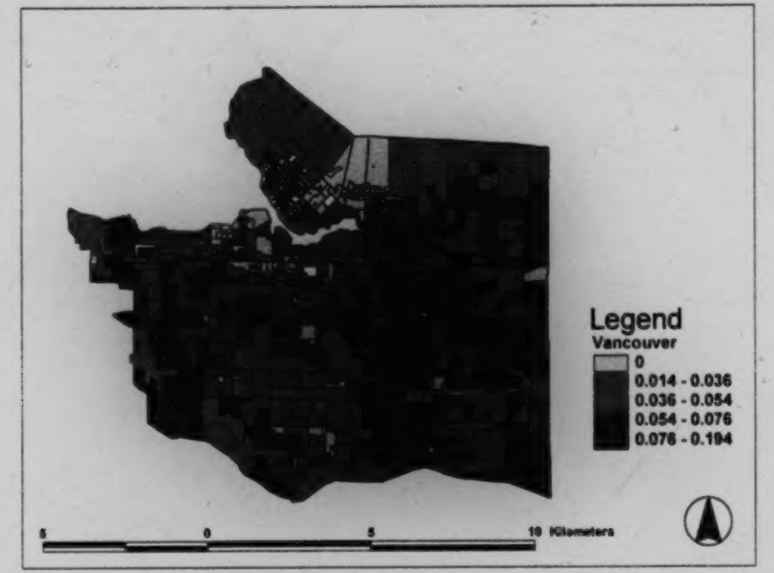
Children's population per capita in Toronto.


Legend
Total Population
1 - 285
285 - 482
482 - 698
698 - 915
915 - 1419
Toronto_boundary.shp
5
0
5
10 Kilometers

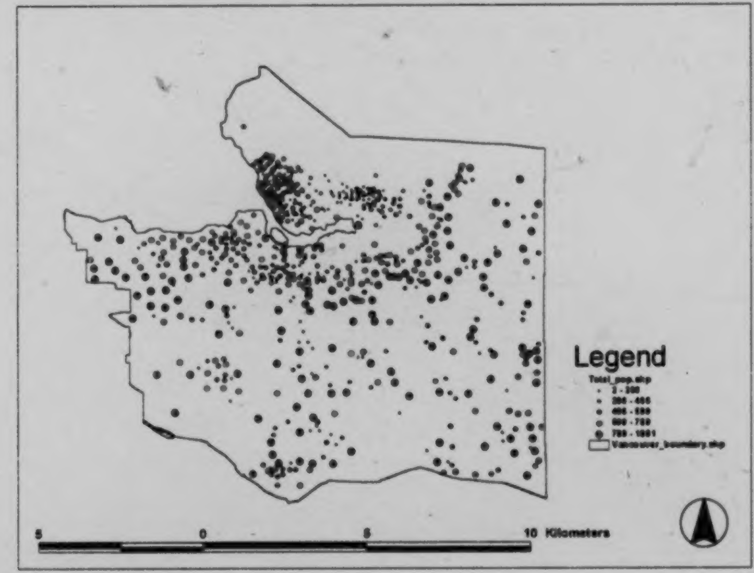
Total population in Toronto.


Legend
Vancouver
0 - 0.068
0.068 - 0.1
0.101 - 0.125
0.125 - 0.175
0.175 - 1.098
5
0
5
10 Kilometers

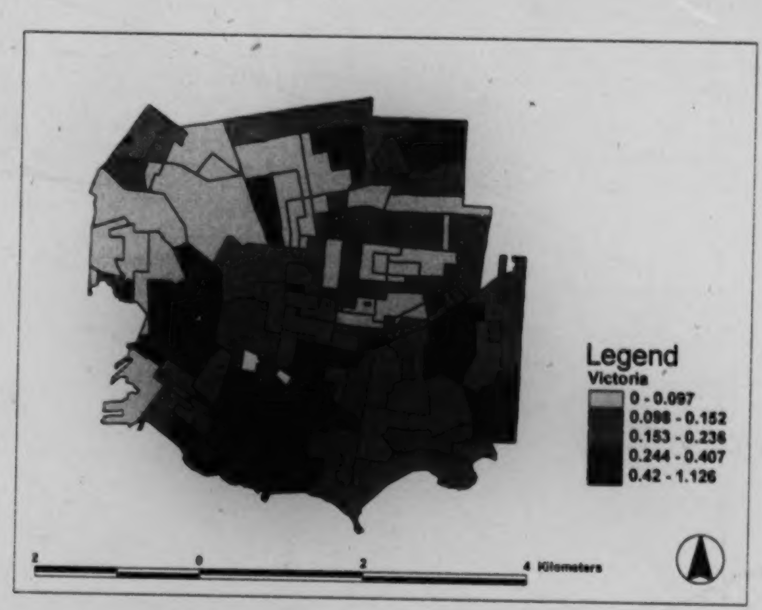
Senior's population per capita in Vancouver.


Legend
Vancouver
0
0.014 - 0.036
0.036 - 0.054
0.054 - 0.076
0.076 - 0.194
5
0
5
10 Kilometers

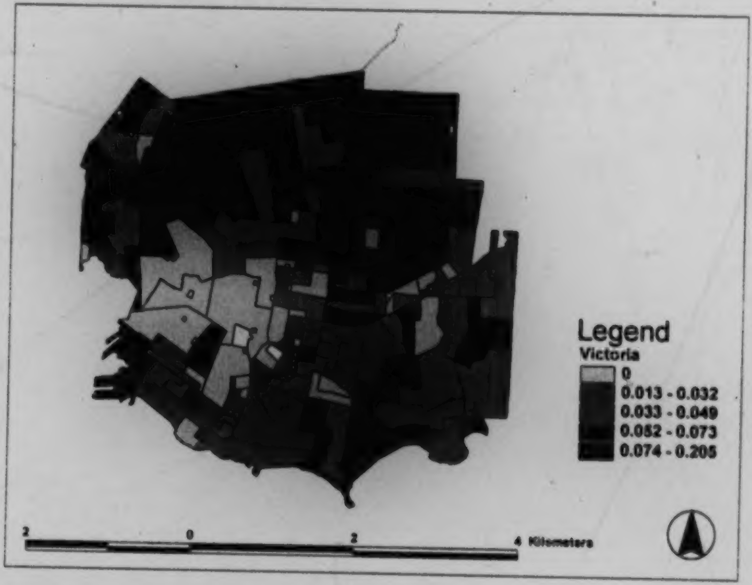
Children's population per capita in Vancouver.


Legend
Total_pop.shp
Vancouver_boundary.shp
5
0
5
10 Kilometers

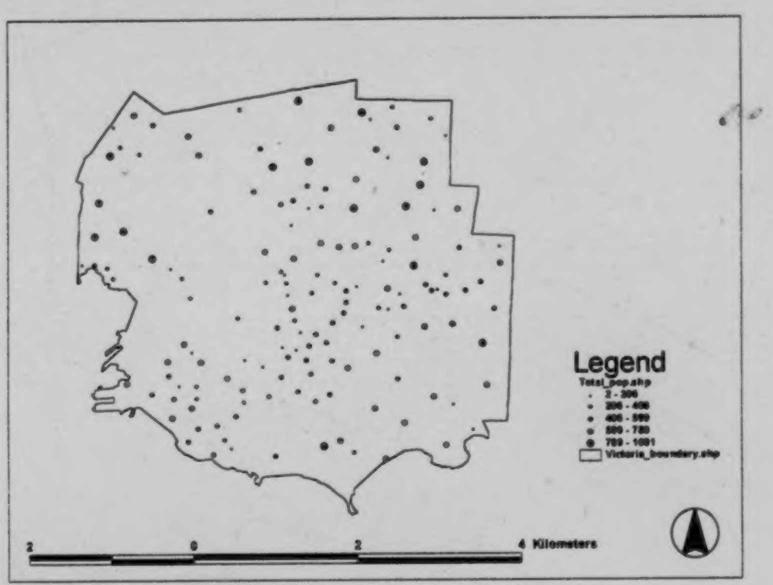
Total population in Vancouver.


Legend
Victoria
0 - 0.097
0.098 - 0.152
0.153 - 0.236
0.244 - 0.407
0.42 - 1.126
2
0
2
4 Kilometers

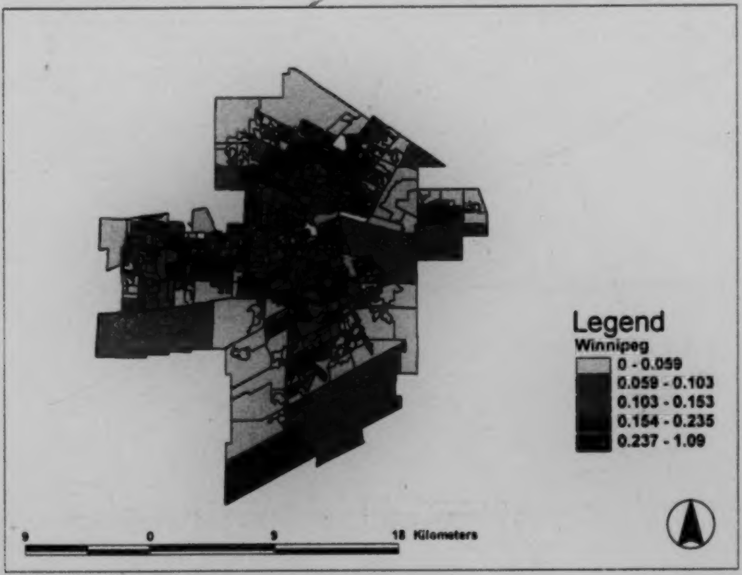
Senior's population per capita in Victoria.


Legend
Victoria
0
0.013 - 0.032
0.033 - 0.049
0.052 - 0.073
0.074 - 0.205
2
0
2
4 Kilometers

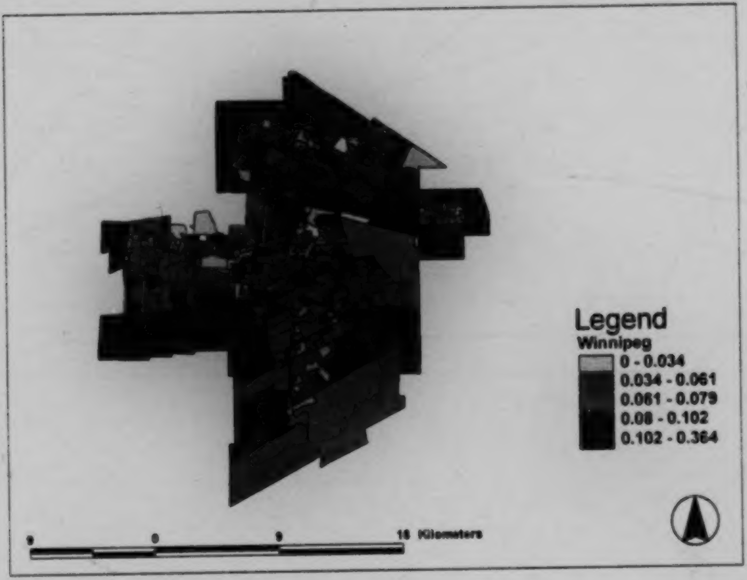
Children's population per capita in Victoria.


Legend
Total_pop.shp
2 - 206
206 - 406
406 - 599
599 - 789
789 - 1081
Victoria_boundary.shp
2
0
2
4 Kilometers

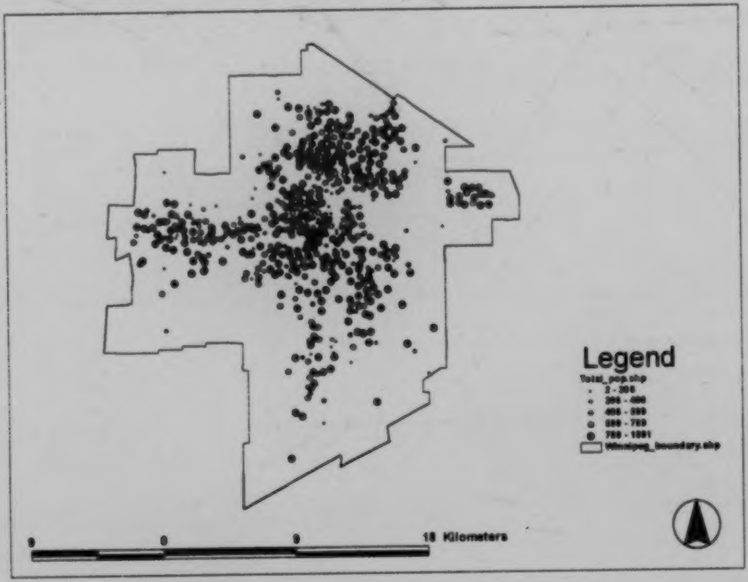
Total population in Victoria.


Legend
Winnipeg
0 - 0.059
0.059 - 0.103
0.103 - 0.153
0.154 - 0.235
0.237 - 1.09
9
0
9
18 Kilometers

Senior's population per capita in Winnipeg.


Legend
Winnipeg
0 - 0.034
0.034 - 0.061
0.061 - 0.079
0.08 - 0.102
0.102 - 0.364
9
0
9
18 Kilometers

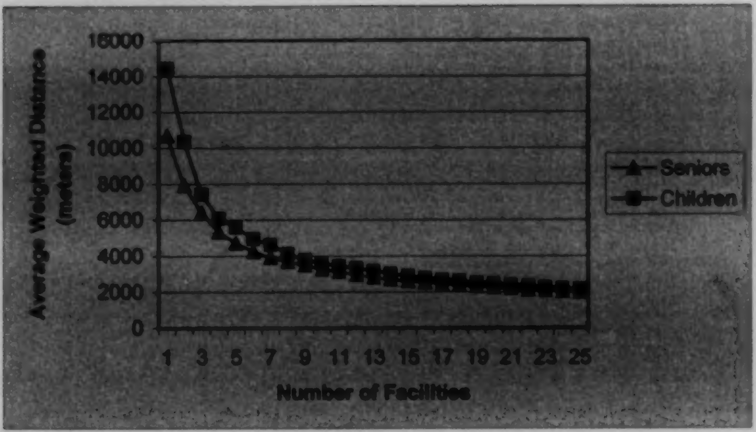
Children's population per capita in Winnipeg.


Legend
Total_pop.shp
Winnipeg_boundary.shp
9
0
9
18 Kilometers

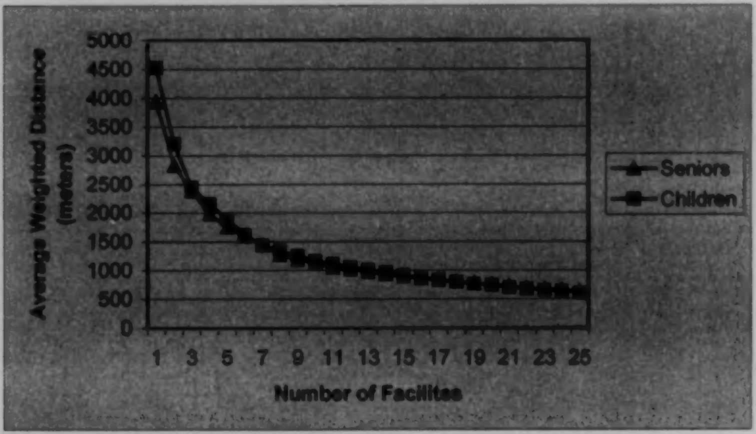
Total population in Winnipeg.

Appendix 6

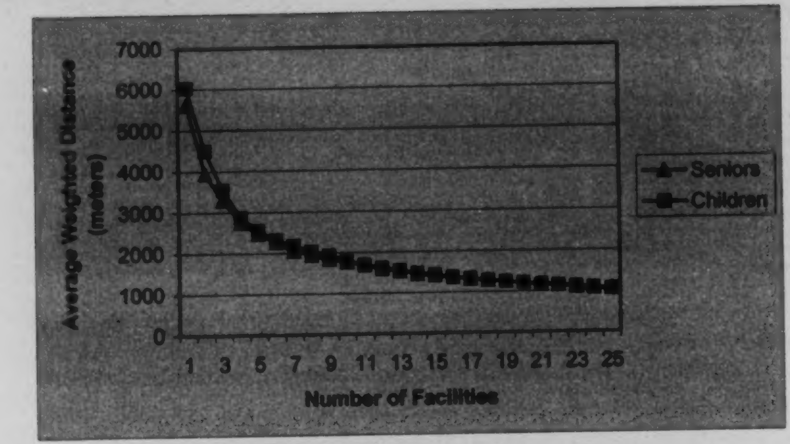
Cost-Effectiveness Graphs



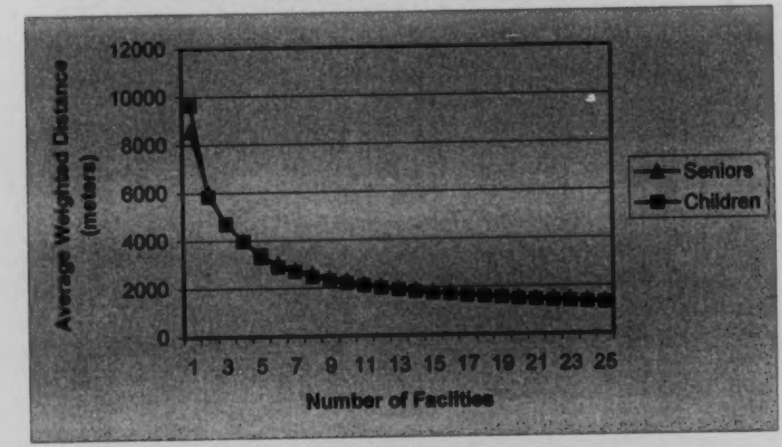
Cost-Effectiveness Graph for Calgary.



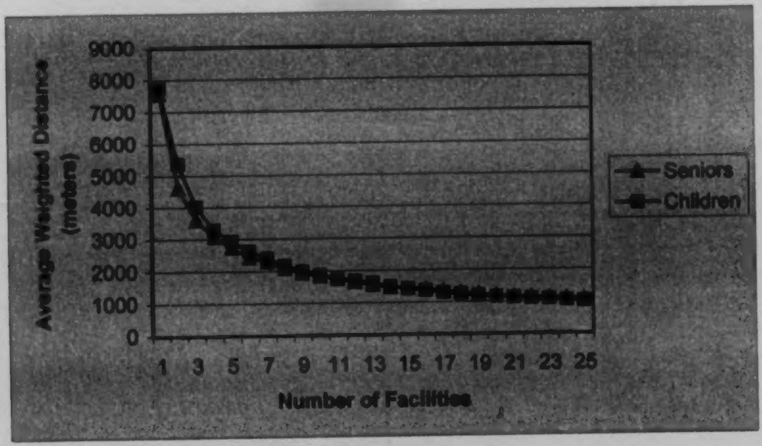
Cost-Effectiveness Graph for Halifax.



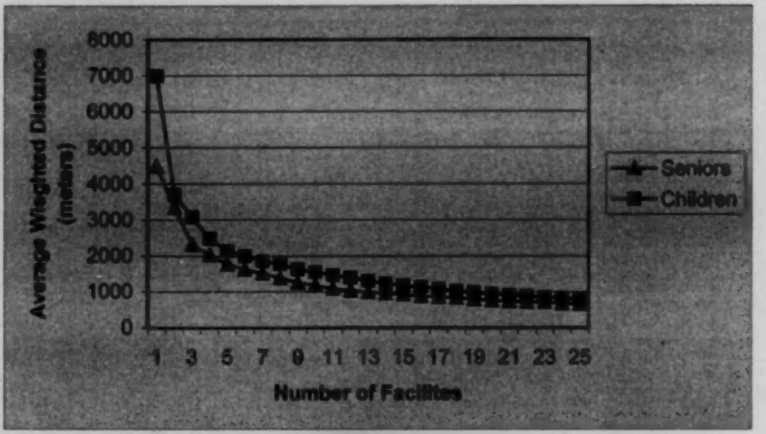
Cost-Effectiveness Graph for Hamilton.



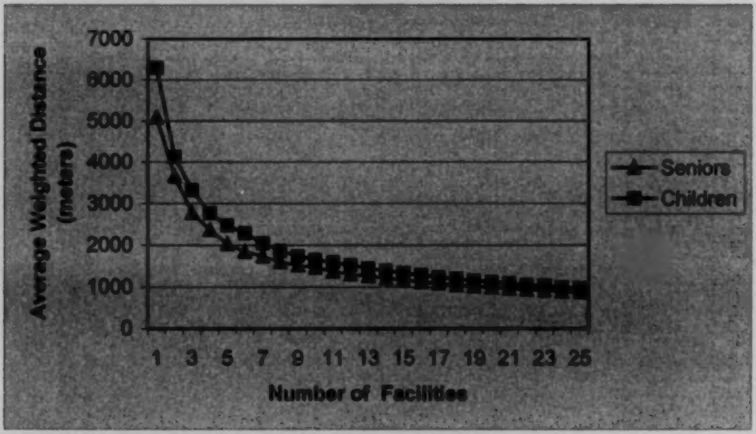
Cost-Effectiveness Graph for North York.



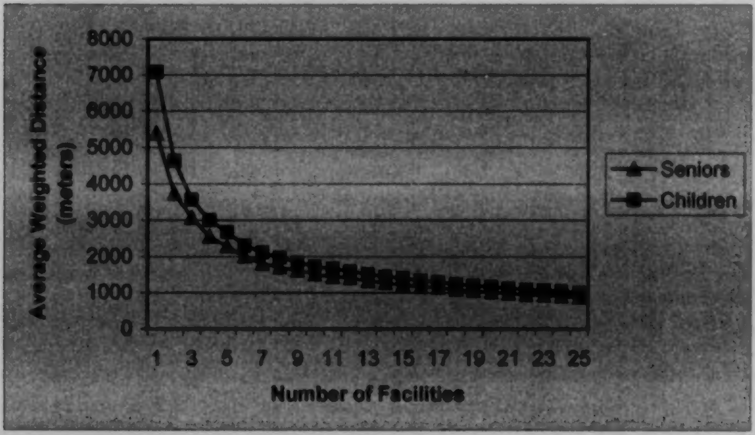
Cost-Effectiveness Graph for Ottawa.



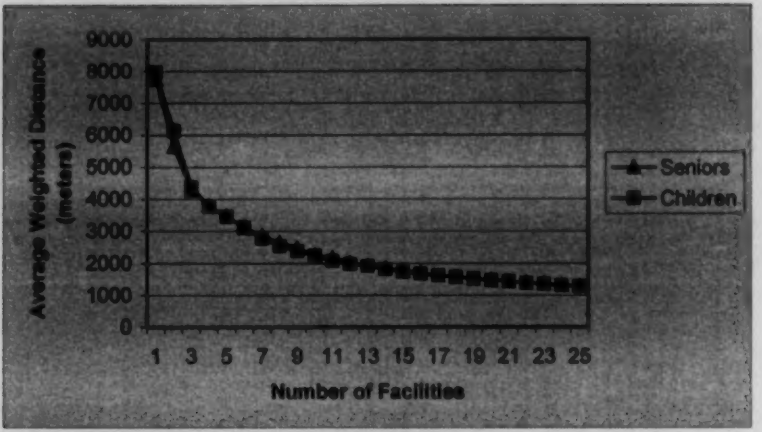
Cost-Effectiveness Graph for Quebec.



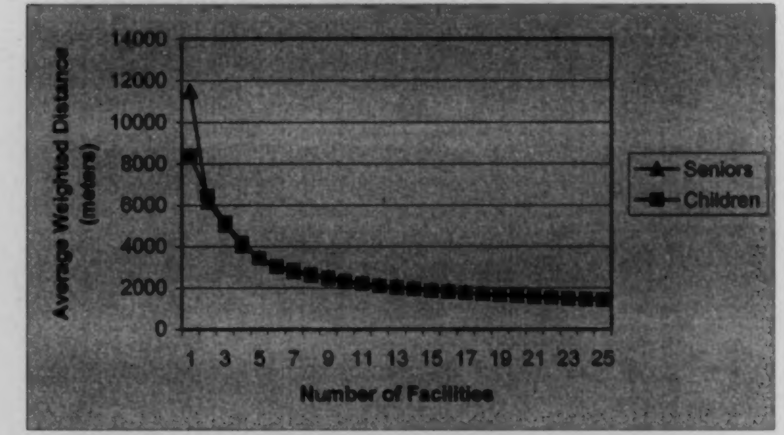
Cost-Effectiveness Graph for Regina.



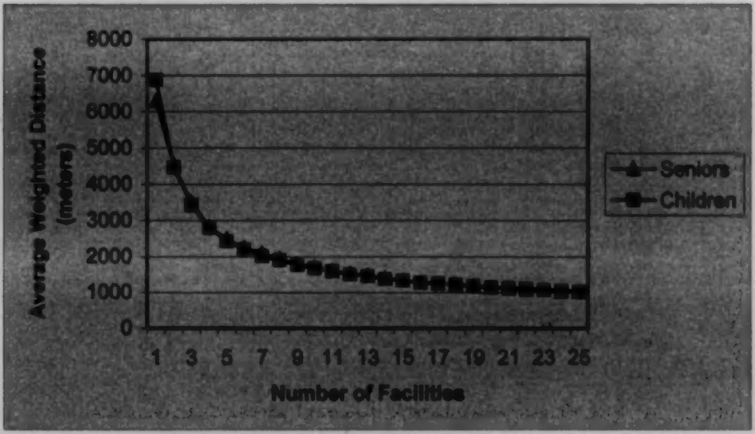
Cost-Effectiveness Graph for Saskatoon.



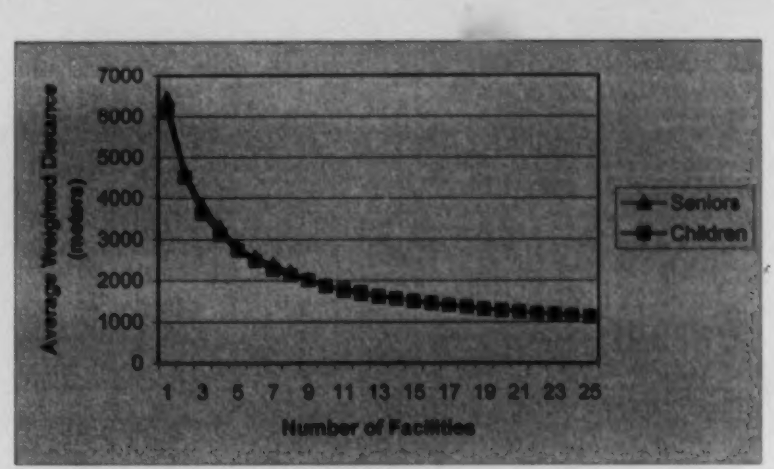
Cost-Effectiveness Graph for Scarborough.



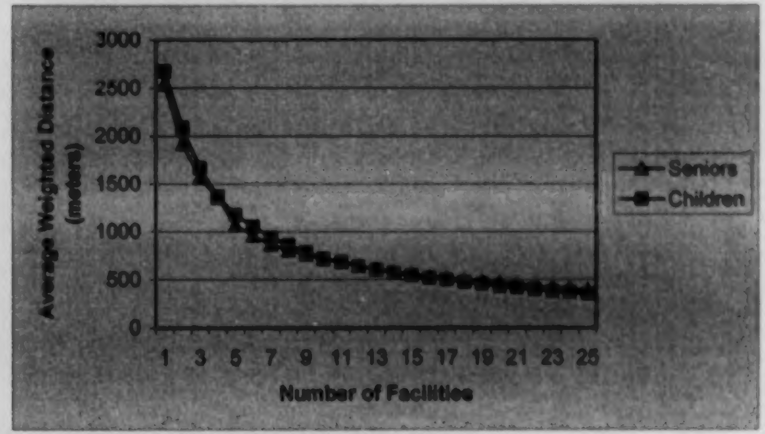
Cost-Effectiveness Graph for Surrey.



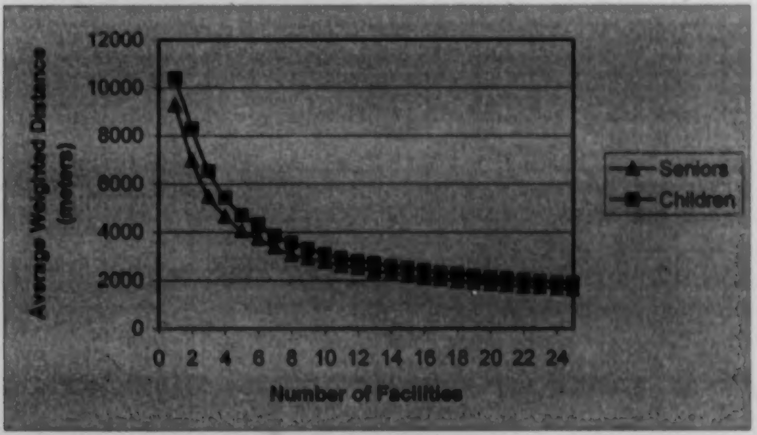
Cost-Effectiveness Graph for Toronto.



Cost-Effectiveness Graph for Vancouver.



Cost-Effectiveness Graph for Victoria.



Cost-Effectiveness Graph for Winnipeg.

Appendix 7

Surrogation Error Results

City / # of Facilities	1	2	3	4	5	6	7	8	9	10	11	12
Calgary	0.17	0.38	0.67	1.62	2.81	2.40	0.96	0.97	1.83	1.70	2.14	1.72
Edmonton	0.15	1.18	1.57	0.68	0.75	1.10	0.79	2.36	1.25	0.65	0.57	1.82
Halifax	0.00	1.38	1.39	1.46	0.83	6.84	2.33	1.46	1.69	2.40	2.30	3.94
Hamilton	0.22	1.33	0.62	1.48	1.94	4.72	1.59	3.27	1.53	1.91	1.10	0.87
North York	1.15	1.09	0.92	1.76	1.17	0.69	0.52	0.99	3.28	2.50	4.24	1.89
Ottawa	1.40	2.93	1.95	4.13	2.30	1.98	1.76	2.85	5.66	6.08	6.65	0.66
Quebec	5.33	0.64	0.14	0.06	1.51	2.90	3.91	0.64	8.32	3.51	4.84	2.25
Regina	0.87	2.37	1.50	1.12	4.09	1.74	4.36	3.63	0.61	0.59	3.45	2.37
Saskatoon	1.06	0.26	0.52	0.50	1.03	1.94	6.12	2.84	3.12	1.63	1.47	2.85
Scarborough	0.27	0.39	0.06	0.09	1.81	0.91	0.12	0.44	0.82	0.37	0.78	0.08
Surrey	0.00	3.87	1.98	0.20	0.49	7.37	0.98	0.69	0.01	0.34	4.07	4.24
Toronto	0.04	1.11	0.92	1.46	2.15	2.51	4.74	2.62	2.51	3.39	4.67	3.73
Vancouver	3.64	1.85	2.35	9.61	6.64	5.59	1.88	4.21	3.68	5.25	2.33	3.66
Victoria	4.98	11.00	6.85	3.37	6.32	6.34	5.07	5.68	7.11	5.67	9.50	5.84
Winnipeg	0.07	0.21	0.36	2.53	0.46	0.25	3.92	1.71	1.22	2.58	2.09	2.07

Surrogation Error Children

City / # of Facilities	13	14	15	16	17	18	19	20	21	22	23	24	25
Calgary	3.40	2.28	4.87	2.46	5.55	2.21	2.30	2.28	1.68	2.06	3.23	2.66	1.53
Edmonton	1.85	0.57	1.98	5.54	4.39	5.10	0.81	1.31	1.02	1.45	1.88	2.01	2.15
Halifax	4.51	8.73	8.12	5.75	5.61	5.53	4.79	3.16	12.3	9.70	12.9	10.3	9.22
Hamilton	0.92	2.97	4.14	2.91	1.59	3.31	1.89	4.46	2.42	2.41	4.52	1.60	4.94
North York	0.70	1.66	1.97	2.92	2.12	1.95	3.48	1.47	3.46	7.01	4.34	2.27	6.06
Ottawa	0.77	3.30	2.60	3.07	5.62	5.96	5.68	8.91	8.10	6.24	3.82	5.50	7.15
Quebec	2.89	6.21	4.64	1.79	5.07	5.64	10.2	11.6	6.55	4.20	8.49	7.90	12.1
Regina	3.32	2.41	3.56	4.39	3.67	0.85	5.33	5.18	6.01	7.39	4.98	7.55	6.32
Saskatoon	4.70	4.63	2.63	4.26	5.24	7.90	4.83	3.66	6.36	7.21	5.28	5.92	4.43
Scarborough	0.73	0.39	0.63	1.20	2.29	1.64	2.63	4.25	1.10	2.76	1.66	2.74	4.51
Surrey	1.97	1.71	3.03	1.49	2.28	2.32	2.96	1.08	2.01	6.25	2.78	3.13	2.89
Toronto	5.08	3.33	2.73	4.09	4.07	4.96	3.73	3.67	5.24	3.12	3.04	2.83	4.05
Vancouver	2.67	3.71	4.57	3.45	3.78	1.67	4.83	5.72	3.58	2.60	3.07	2.79	4.24
Victoria	12.2	12.1	13.2	11.1	18.0	11.3	9.97	11.5	14.6	15.2	15.8	14.9	19.2
Winnipeg	1.30	1.18	1.76	0.72	1.94	1.54	2.84	4.77	2.94	2.00	1.58	3.34	2.25

Surrogation Error Children (con't)

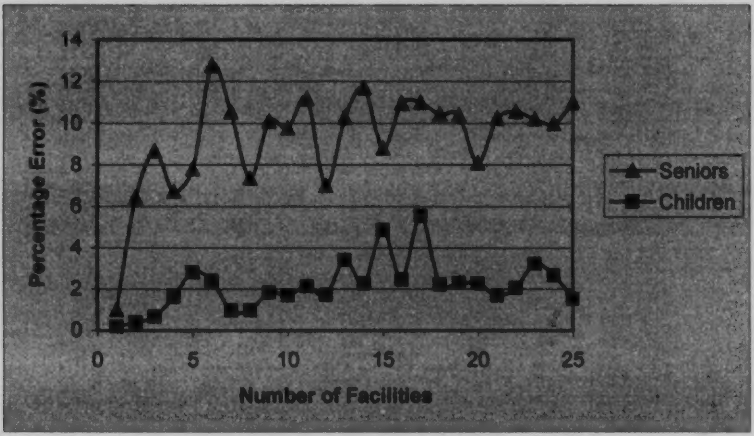
City / # of Facilities	1	2	3	4	5	6	7	8	9	10	11	12
Calgary	0.97	6.40	8.66	6.72	7.80	12.76	10.53	7.37	10.10	9.75	11.17	7.00
Edmonton	0.18	3.68	9.15	8.09	13.94	10.99	8.14	9.83	7.88	8.19	5.91	6.58
Halifax	0.00	0.87	1.96	8.92	6.51	6.44	4.75	3.26	3.22	6.01	6.09	9.72
Hamilton	0.30	0.00	5.75	5.53	4.91	6.99	7.53	5.46	4.69	3.14	6.53	8.25
North York	0.12	3.24	2.41	0.62	4.83	1.98	1.83	2.19	2.57	4.62	4.32	3.79
Ottawa	0.00	2.00	1.60	3.12	3.62	6.26	6.98	7.86	5.98	4.54	5.22	10.84
Quebec	2.07	4.01	3.55	2.46	11.25	13.18	9.37	10.07	9.71	14.56	7.09	11.89
Regina	5.20	5.99	12.85	21.86	7.88	8.31	8.85	14.56	12.83	13.16	10.70	12.03
Saskatoon	0.53	14.07	11.17	13.38	13.09	13.92	14.82	23.52	14.59	21.16	22.23	10.51
Scarborough	1.56	2.37	2.43	3.43	1.25	7.07	4.25	1.79	3.34	3.25	1.47	1.83
Surrey	2.00	1.68	2.22	1.39	1.27	2.12	3.83	5.14	4.28	7.18	3.78	5.37
Toronto	0.33	1.11	0.23	0.41	1.21	1.20	1.28	2.21	1.09	2.00	3.08	4.25
Vancouver	0.13	0.18	0.38	1.05	1.41	1.18	1.54	2.08	4.22	6.37	11.20	3.60
Victoria	3.43	4.95	0.55	1.51	6.47	7.48	8.88	5.50	5.85	10.18	7.22	12.12
Winnipeg	0.00	12.40	7.30	2.76	3.01	4.65	3.57	4.88	4.91	4.96	6.65	4.34

Surrogation Error Seniors

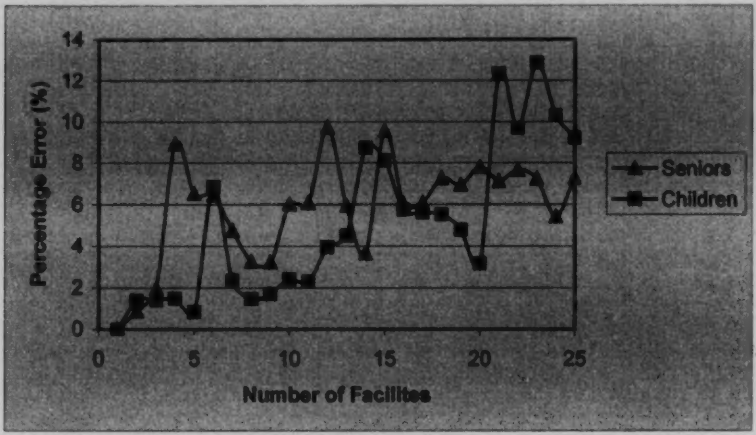
City / # of Facilities	13	14	15	16	17	18	19	20	21	22	23	24	25
Calgary	10.3	11.7	8.78	11.0	11.0	10.4	10.4	8.09	10.2	10.6	10.2	9.97	11.0
Edmonton	7.92	11.9	10.3	9.74	7.08	8.17	10.5	8.35	7.33	5.45	6.25	5.96	6.96
Halifax	5.93	3.65	9.57	6.03	6.09	7.26	6.94	7.81	7.14	7.66	7.24	5.42	7.29
Hamilton	8.89	6.98	8.97	10.9	7.13	7.36	6.24	6.15	7.63	9.17	8.53	11.7	9.26
North York	6.34	4.58	5.06	8.76	9.42	7.53	9.33	7.68	6.65	5.18	9.97	6.90	6.37
Ottawa	9.37	8.43	7.70	6.48	4.33	4.04	6.54	10.2	8.89	5.46	8.33	7.03	6.08
Quebec	13.7	10.1	7.82	12.5	6.22	11.7	11.6	11.7	15.5	19.7	17.1	17.5	16.6
Regina	21.4	17.8	18.6	17.1	17.8	18.2	19.9	18.3	21.7	23.9	24.9	19.2	26.8
Saskatoon	10.2	18.0	21.0	21.7	19.3	15.7	25.7	20.7	22.1	18.1	24.5	26.6	28.7
Scarborough	3.08	1.16	3.25	4.58	5.43	8.24	9.09	8.68	9.06	10.5	6.74	8.85	8.86
Surrey	12.8	8.58	6.92	9.96	9.67	12.4	11.3	12.3	12.3	10.2	10.0	12.2	13.1
Toronto	0.99	5.04	2.03	3.69	4.09	4.00	3.19	4.26	3.39	3.96	2.63	4.38	1.81
Vancouver	3.35	2.61	4.72	2.31	1.40	3.10	3.97	5.88	5.22	5.59	7.08	6.48	6.46
Victoria	7.23	9.07	12.1	14.7	12.5	17.0	18.1	17.9	13.8	15.7	17.8	19.7	18.8
Winnipeg	7.09	5.94	7.71	6.67	8.85	11.0	10.9	8.23	8.65	10.1	10.2	10.9	12.1

Surrogation Error Seniors (con't)

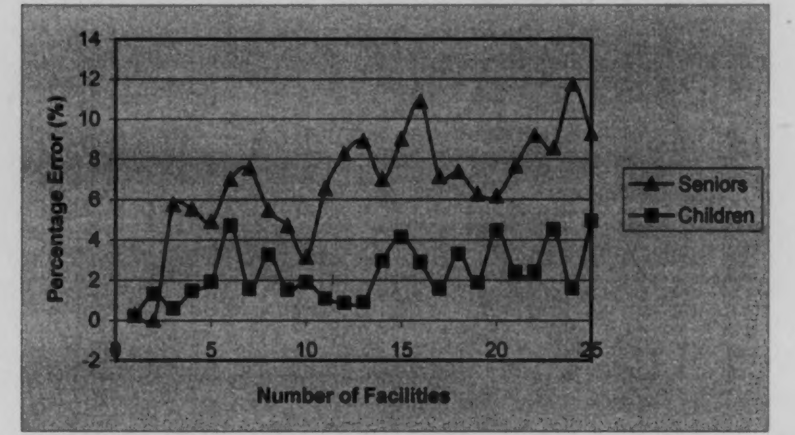
Surrogation Error Graphs - Seniors and Children compared to the Total Population (surrogate data)



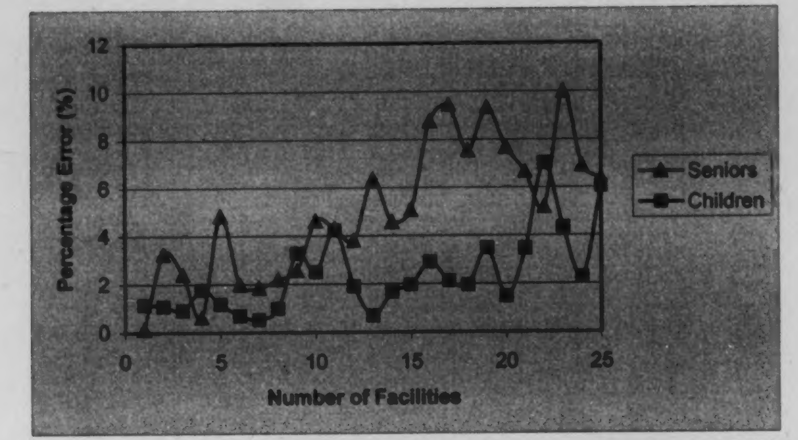
Surrogation Error - children and seniors demographic groups compared to the total population (surrogate data) for Calgary.



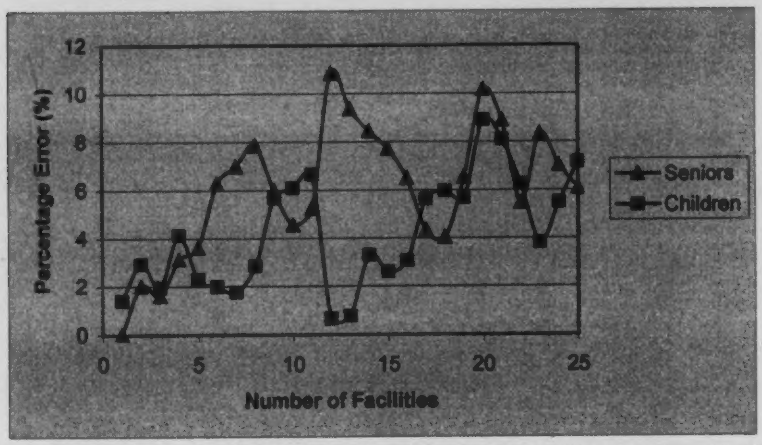
Surrogation Error - children and seniors demographic groups compared to the total population (surrogate data) for Halifax.



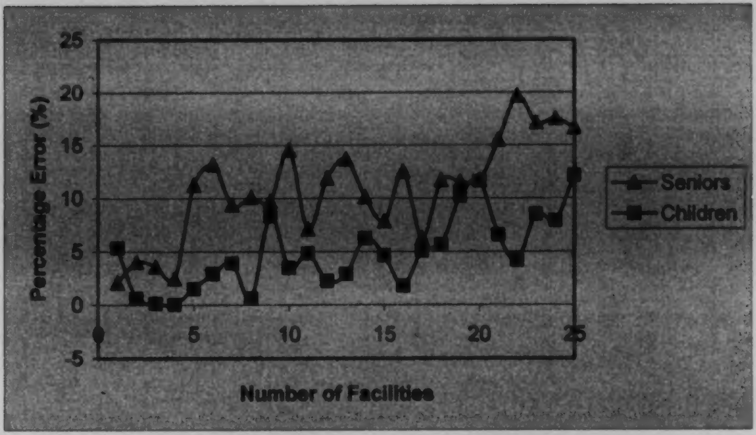
Surrogation Error - children and seniors demographic groups compared to the total population (surrogate data) for Hamilton.



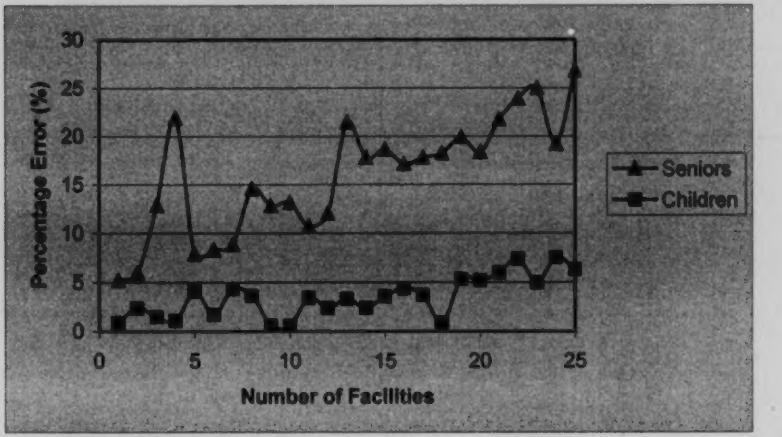
Surrogation Error - children and seniors demographic groups compared to the total population (surrogate data) for North York.



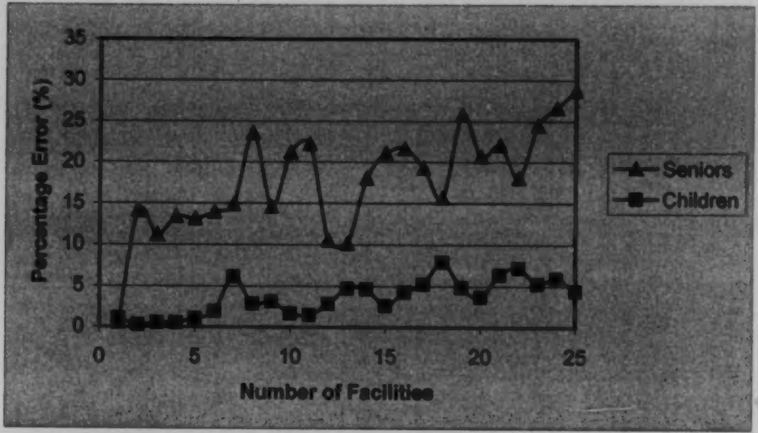
Surrogation Error - children and seniors demographic groups compared to the total population (surrogate data) for Ottawa.



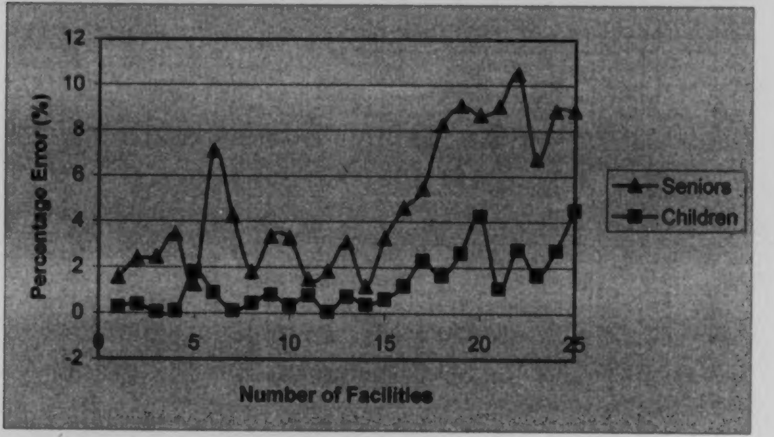
Surrogation Error - children and seniors demographic groups compared to the total population (surrogate data) for Quebec.



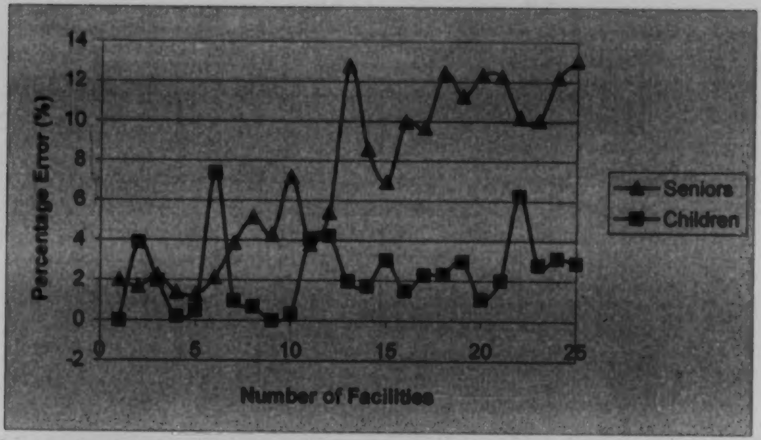
Surrogation Error - children and seniors demographic groups compared to the total population (surrogate data) for Regina.



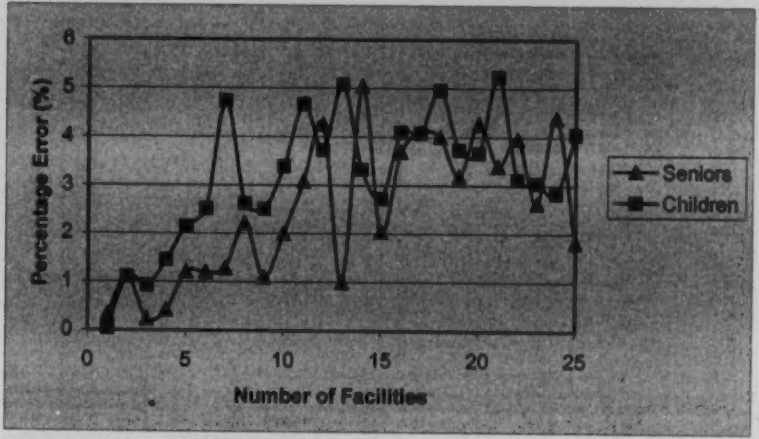
Surrogation Error - children and seniors demographic groups compared to the total population (surrogate data) for Saskatoon.



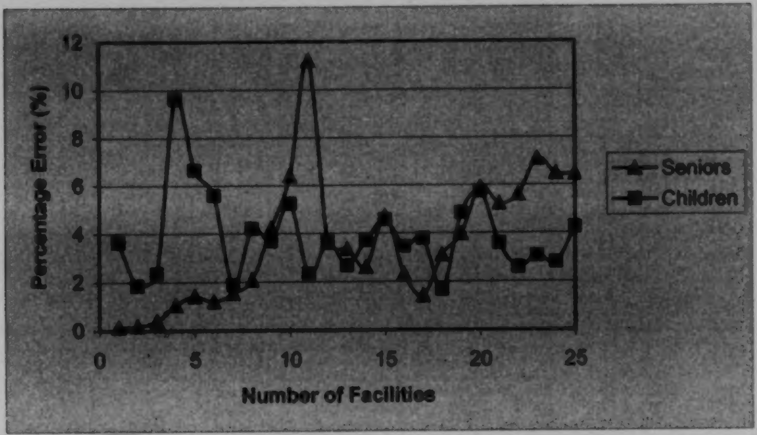
Surrogation Error - children and seniors demographic groups compared to the total population (surrogate data) for Scarborough.



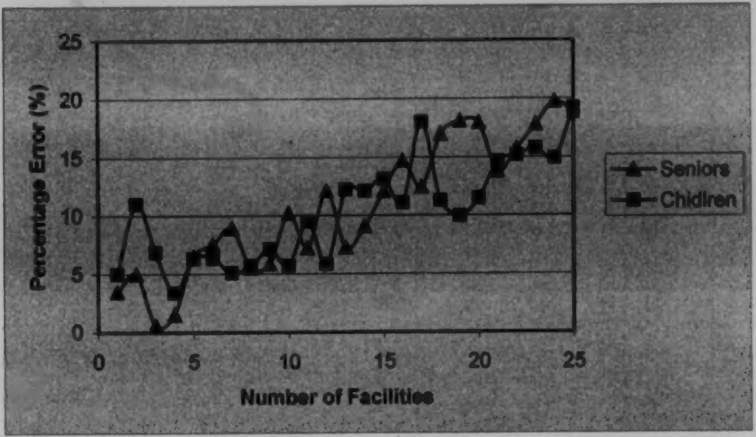
Surrogation Error - children and seniors demographic groups compared to the total population (surrogate data) for Surrey.



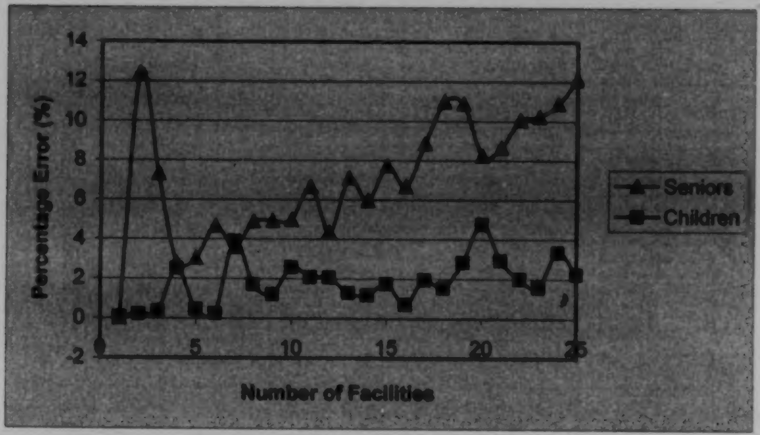
Surrogation Error - children and seniors demographic groups compared to the total population (surrogate data) for Toronto.



Surrogation Error - children and seniors demographic groups compared to the total population (surrogate data) for Vancouver.



Surrogation Error - children and seniors demographic groups compared to the total population (surrogate data) for Victoria.



Surrogation Error - children and seniors demographic groups compared to the total population (surrogate data) for Winnipeg.

Appendix 8

ArcView® 3.1 Avenue® Computer Code used to create the ARCLA extension

```
' Name:  A R C L A - Source Code
'
' Title: ArcView's Location-Allocation Extension - Source Code
'
' Topics:  Location-Allocation, Geo-demographic analysis
'
' Description:   This script is one part of three required scripts needed to create the ARCLA extension.  The user  needs
to
'                 modify ArcView's Interface by adding custom menus, buttons, etc and then save these modifications as
'                 a Project and then run this script while ensuring that the ARCLA Install and ARCLA Uninstall scripts
'                 presenting the project.  This extension was created to be used with the 1996 Statistics Canada Data set
for
'                 Location-Allocation Modelling.
'
' Requires:    Data sets from the 1996 Statistics Canada Survey
'
' Self:
'
' Returns:  The ARCLA extension (arcla.avx) file
'-----------------------------
'the Filename of the extension
'
theExtensionFile = "$HOME\arcla.avx"
'-----------------------------
 'The Name of the extension to be displayed in the
 'extension dialog
theExtensionName = "A R C L A"
'-----------------------------
 'The description of the extension to be displayed
 ' in the extension dialog
 theDescription = "A R C L A - ArcView's Location-Allocation Modeling extension.  This extension has been created
as a Tool for Location Researchers and should be used primarily with 1996 Census Canada Data sets."
'-----------------------------
 'The version of the extension
theVersion = 1.0
'-----------------------------
 'Install Script
 ' The Script to use to install the extension
 'NOTE: "My Extension Install" is a pre-perpared script for this
InScriptName="ARCLA Install"
'-----------------------------
 'Uninstall Script
 ' The Script to use to uninstall the extension
 'NOTE: "My Extension Uninstall" is a pre-perpared script for this
UnScriptName="ARCLA Uninstall"
'-----------------------------
 'Any document Names to include
 ' This is a list of any documents to include
 ' e.g. theDocs={"View1","Layout1"}
theDocs={}
'-----------------------------
 'Controls (Buttons and normal Tools) to include
 'Controls {GUI name,ControlType,ScriptName}
 'e.g. TheControlList={{"View","ButtonBar","View.ZoomIn"}, {"View","ToolBar","View.Identify"}}
TheControllist={}
'-----------------------------
 'Tool menus to include
```

```
' This is a list of which tool menus to include.  Tool menus are specified by giving a GUI name and a script name of
one of
' the tools in the tool menu Using this to locate the tool menu ALL other tools inb that menu and their scripts will be
' extracted theToolMenu={{"View","View.PointTool"}}
TheToolMenuList={}
'----------------------------
'The Menus to include (a list of menu items listing the doc, the top menu, and the script for the menu item)
'The MenuList {Doc name, Main Menu name , menu item Scriptname}
' e.g. The MenuList={{"View", "File", "View.Export"}}
TheMenuList={{"Project", "LA Data Preparation", "View.NewView"},
        {"Project", "LA Data Preparation", "Space"},
        {"Project", "LA Data Preparation", "Query.Census1996Data"},
        {"Project", "LA Data Preparation", "Space"},
        {"Project", "LA Data Preparation", "Project.SetProjectWorkDir"},
        {"View", "Query Census Database","Query.Census1996Data"},
        {"View", "LA Data Preparation","Table.ConvertCensus2EventTheme"},
        {"View", "LA Data Preparation","View.AddXYCoordToFTab"},
        {"View", "LA Data Preparation","Space"},
        {"View", "LA Data Preparation","Table.QueryNonZero"},
        {"View", "LA Data Preparation","Space"},
        {"View", "LA Data Preparation","Table.ExportDemandPoints"},
        {"View", "LA Data Preparation","Table.ExportFacilityPoints"},
        {"View", "Solving LA Problems","Project.Create2TextFiles"},
        {"View", "Solving LA Problems","Project.SearchForLATextFiles"},
        {"View", "Solving LA Problems","Space"},
        {"View", "Solving LA Problems","Project.StartLA"},
        {"View", "Mapping LA Solutions","Project.ImportFacilitiesSolutionFile"},
        {"View", "Mapping LA Solutions","Table.QueryFacilityShapefile"},
        {"View", "Mapping LA Solutions","Space"},
        {"View", "Mapping LA Solutions","Project.ImportFinalDemandsFile"},
        {"View", "Mapping LA Solutions","Table.QueryDemandShapefile"},
        {"View", "Mapping LA Solutions","Space"},
        {"View", "Mapping LA Solutions","Theme.ConvertEAuIDtoNumber"},
        {"View", "Mapping LA Solutions","Space"},
        {"View", "Mapping LA Solutions","View.LocationAllocationDiagram"},
        {"View", "Mapping LA Solutions","Space"},
        {"View", "Mapping LA Solutions","Theme.LegendFacilityUniqueSymbol"},
        {"View", "Mapping LA Solutions","Theme.GraduatedSymbolMap"},
        {"View", "Mapping LA Solutions","Space"},
        {"View", "Mapping LA Solutions","Theme.ChoroplethMap"},
        {"View", "Mapping LA Solutions","View.CalculatePolygonGeometry"},
        {"View", "Mapping LA Solutions","Space"},
        {"View", "Mapping LA Solutions","Theme.LegendMakeSimple"},
        {"View", "Utilities","View.RandomSelection"},
        {"View", "Utilities","Space"},
        {"View", "Utilities","Table.DisplaySelectedFields"},
        {"View", "Utilities","Space"},
        {"View", "Utilities","View.HideLegendForInvisibleThemes"},
        {"View", "Utilities","Space"},
        {"View", "Utilities","View.DisplaySelectedThemes"},
        {"View", "Utilities","Space"},
        {"View", "Utilities","View.SetLambertProjection"},
        {"Table","Utilities","Table.DisplaySelectedFields"},
        {"Table","Utilities","Table.FindDuplicatesValues"} }
'----------------------------
'The scripts to include, not in controls or menus
'e.g. theScripts={"My.Script","View.export"}
theScripts={}
'----------------------------
'The  Dependencies
theDependencies={}
'----------------------------
```

```
' ATTENTION!!!!!!!!!!!!!!!!
' Do NOT alter the script after these lines, the remainer of this script uses the lists you provided above to build the
' extension object.
'
________________________________________________________________
'Create a total list of the needed scripts
TheNeeded={}
for each ControlScript in thecontrolList
 TheNeeded.add(ConTrolScript.get(2))
end
For each MenuScript in theMenuList
 theNeeded.add(MenuScript.get(2))
end
totalscripts=thescripts.merge(theNeeded)
TotalScripts.removeduplicates
theInstall=av.GetProject.FindScript(InscriptName)
if (theInstall=NIL) then
  Msgbox.Error("The install Script "+InScriptName+" was not found","Script not found")
  return(nil)
end
theUninstall=av.GetProject.FindScript(unscriptName)
if (theUninstall=NIL) then
  Msgbox.Error("The install Script "+UnScriptName+" was not found","Script not found")
  return(nil)
end
'Create the extension
' Extension.Make(ExtensionFile, theExtensionName, InstallScript, UninstallScript, Dependancies_
myExt = Extension.Make(theExtensionFile.asFilename,
            theExtensionName,
            theInstall,
            theUninstall,
            theDependencies)
TheDocList={}
TheTotalControls={}
ThetotalMenus={}
theTotalToolMenus={}
' Process the Document List
for each aDoc in TheDocs
  if ((av.getproject.findDoc(aDoc)="Null").NOT) then
    TheDocList.Add(av.GetProject.FindDoc(aDoc))
   else
    MsgBox.Warning("The Doc "+adoc.asstring+" cannot be found in the current project.","Script Error")
    return(nil)
 end
end
'Add the List of Documents
MyExt.add(TheDocList)
' Process the control List
'
____________________________________________

'Controls {GUI,ControlType,Scriptname}
'
____________________________________________
for each aControl in TheControlList
  theControlDoc=av.getproject.findGUI(aControl.get(0))
   if (theControlDoc=NIL) then
    MsgBox.Warning("The GUI "+aControl.get(0)+" cannot be found in the current project.","Script Error")
    return(nil)
   end
  thecommand="av.getproject.findGUI("""+aControl.get(0)+""").Get"+acontrol.get(1)
  'msgbox.info(thecommand,"")
  thescript1=Script.Make(thecommand)
  thecontrolset=thescript1.doit("")
  theFoundControl=TheControlSet.FindbyScript(aControl.get(2))
```

```
    if (theFoundControl=NIL) then
     MsgBox.Warning("No Control found with the script "+aControl.get(2)+" in "+acontrol.get(1),"Warning")
     return(nil)
    end
    if (theFoundControl.is(ToolMenu)) then
     MsgBox.Warning("The script "+aControl.get(2)+" in "+acontrol.get(1)+" is for a tool menu.","Not Supported")
     return(nil)
    end

  TheTotalControls.Add({aControl,thefoundControl,theControlSet.getcontrols.find(thefoundcontrol)})
end
'Add to Ext
MyExt.add(TheTotalControls)
'__________________________________________

'The MenuList {Doc, Menu, MenuScript}
'__________________________________________
themenuitemidx = 1
For each aMenu in themenulist
  mDoc=aMenu.get(0)
  mMenu=aMenu.get(1)
  mScript=aMenu.get(2)
  themDoc=av.getproject.findGUI(mDoc)
  if (themDoc=NIL) then
     MsgBox.Warning("The GUI "+mDoc+" cannot be found in the current project.","Script Error")
     return(nil)
   end
  theMbar=av.getproject.findGUI(mDoc).GetMenuBar
  themenu=theMbar.findbylabel(mMenu)
  if (themenu=NiL) then
     MsgBox.Warning("The menu named "+mMenu+" is not here.","Script Error")
     return(nil)
  end
  themenucontrol=themenu.findbyScript(mScript)
  if (themenucontrol=NiL) then
     if(mScript.AsString = "Space") then
       themenucontrol = Space.Make
       themenuitemidx = themenuitemidx + 1
     else
       MsgBox.Warning("The script "+mScript+" not found.","Warning")
       return(nil)
     end
  else
  themenuitemidx=themenu.getcontrols.find(themenucontrol)
  end
  thetotalmenus.add({amenu,theMenuControl,theMenuItemIdx})
end
myext.add(thetotalmenus)
'__________________________________________
'Process the Tool Menu List
'__________________________________________
for eacn aControl in TheToolMenuList
  theControlDoc=av.getproject.findGUI(aControl.get(0))
   if (theControlDoc=NIL) then
     MsgBox.Warning("The GUI "+aControl.get(0)+" cannot be found in the current project.","Script Error")
     return(nil)
   end
  thecontrolset=theControlDoc.getToolbar
  theFoundControl=TheControlSet.FindbyScript(aControl.get(1))
   if (theFoundControl=NIL) then
     MsgBox.Warning("No Control found with the script "+aControl.get(1)+" in "+acontrol.get(1),"Warning")
     return(nil)
   end
   if (theFoundControl.is(ToolMenu).not) then
    MsgBox.Warning("The script "+aControl.get(1)+" is NOT for a tool menu.","User Error")
    return(nil)
   end
   TheTotalToolMenus.Add({aControl,thefoundControl,theControlSet.getcontrols.find(thefoundcontrol)})
end
myext.add(thetotalToolmenus)
'________________________________________
' Process the Script list
'________________________________________
for each ascript in TotalScripts
 if ((av.getproject.findscript(ascript)="Null").NOT) then
   myExt.Add(av.GetProject.FindScript(ascript))
  else
    MsgBox.Warning("The script "+ascript+" cannot be found in the current project.","Script Eror")
    return(nil)
 end
end
 myExt.SetAbout(theDescription)
myExt.SetExtVersion(theVersion)
myExt.Commit
```

```
' Name:  ARCLA Install
'
' Title: Install extension
'
' Topics:
'
' Description:  This script is one part of three required scripts needed to create the           ARCLA extension.
'
' Requires:
'
' Self:
'
' Returns:
'****
'DO NOT EDIT THIS SCRIPT!!!
'****
if (av.getproject=nil) then return(nil) end
theDocs = SELF.get(0)
theControlList = SELF.get(1)
theMenuList = SELF.get(2)
theToolMenuList=SELF.Get(3)
theProject=Av.getproject
'****
'Add the Docs
'****
for each adoc in theDocs
    theProject.addDoc(adoc)
end
'****
'Add the Controls
'****
for each totalControl in theControlList
    'The Control list
    acontrol=totalControl.get(0)
    'The physical control
    theControl = totalControl.get(1)
    'The control Index
    theCindex=totalControl.get(2)
    'Find the DocGUI
    theControlDoc=av.getproject.findGUI(aControl.get(0))
    if (theControlDoc=NIL) then
        MsgBox.Warning("The GUI "+aControl.get(0)+" cannot be found in the current project.","Script Eror")
        return(nil)
    end
    'This finds the control set
    thecommand="av.getproject.findGUI("""+aControl.get(0)+""").Get"+acontrol.get(1)
    thescript1=Script.Make(thecommand)
    thecontrolset=thescript1.doit("")
    'Add the control to the control set
    theControlSet.Add(theControl,theCindex)
end
'****
'Add the menus
'****
for each totalcontrol in theMenuList
    'The Control list
    acontrol=totalControl.get(0)
    mDoc=acontrol.get(0)
    mMenu=acontrol.get(1)
    mMenuItem=acontrol.get(2)
    'The physical control
    theControl = totalControl.get(1)
    'The control Index
    theCindex=totalControl.get(2)
    'Find the DocGUI
    theControlDoc=av.getproject.findGUI(aControl.get(0))
    if (theControlDoc=NIL) then
        MsgBox.Warning("The GUI "+aControl.get(0)+" cannot be found in the current project.","Script Eror")
        return(nil)
    end
    theMbar=av.getproject.findGUI(mDoc).GetMenuBar
    themenu=theMbar.findbylabel(mMenu)
    if (themenu=NiL) then
        themenu=menu.make
        themenu.setlabel(mMenu)
        theMbar.add(themenu,999)
    end
    themenu.add(thecontrol, theCindex)
end
'****
'Add the Tool Menus
'****
for each totalControl in theToolMenuList
    'The Control list
    acontrol=totalControl.get(0)
    'The physical control
    theControl = totalControl.get(1)
    'The control Index
    theCindex=totalControl.get(2)
    'Find the DocGUI
    theControlDoc=av.getproject.findGUI(aControl.get(0))
    if (theControlDoc=NIL) then
        MsgBox.Warning("The GUI "+aControl.get(0)+" cannot be found in the current project.","Script Eror")
        return(nil)
    end
    'This finds the control set
    thecommand="av.getproject.findGUI("""+aControl.get(0)+""").Get"+acontrol.get(1)
    thescript1=Script.Make(thecommand)
    thecontrolset=av.getproject.findGUI(aControl.get(0)).GetToolBar
    'Add the control to the control set
    theControlSet.Add(theControl,theCindex)
end
av.getproject.setmodified(true)
'****
'And the scripts add themselves
'****
```

```
' Name: ARCLA Uninstall
'
' Title: Install extension
'
' Topics:
'
' Description:  This script is one part of three required scripts needed to create the          ARCLA extension.
'
' Requires:
'
' Self:
'
' Returns:
'****
'DO NOT EDIT THIS SCRIPT!!!
'****
'The SELF is the Extension
theDocs = SELF.get(0)
theControlList = SELF.get(1)
theMenuList = SELF.get(2)
theToolMenuList=SELF.get(3)
theProject=Av.getproject
'****
'Add the Docs
'****
for each adoc in theDocs
    If (theProject.finddoc(adoc.getname)<>NIL) then
        theAnswer=msgbox.yesno("Remove the Document "+adoc.getname+"?","Remove Document?",TRUE)
        if (theAnswer=TRUE) then theProject.RemoveDoc(adoc) end
    end
end
'****
'Remove the Controls
'****
for each totalControl in theControlList
    'Get the control list from the Ext
    acontrol=totalControl.get(0)
    'Get the physical Control
    theControl = totalControl.get(1)
    'Get the Controls Index
    theCindex=totalControl.get(2)
    'Find the DocGUI for the Control
    theControlDoc=av.getproject.findGUI(aControl.get(0))
    if (theControlDoc=NIL) then
        MsgBox.Warning("The GUI "+aControl.get(0)+" cannot be found in the current project.","Script Eror")
        return(nil)
    end
    'This sequence finds the appropriate control set
    thecommand="av.getproject.findGUI("""+aControl.get(0)+""").Get"+acontrol.get(1)
    thescript1=Script.Make(thecommand)
    thecontrolset=thescript1.doit("")
    'See if the control is in the set , if so remove it
    if (theControlSet.GetControls.find(theControl)<>NIL) then
        theControlSet.remove(theControl)
        if (thecontrol = "ToolBar") then
            theControlSet.selectdefault
        end
    end
end
'****
'Remove the Menus
'****
```

```
for each totalcontrol in theMenuList
    'The Control list
    acontrol=totalControl.get(0)
    mDoc=acontrol.get(0)
    mMenu=acontrol.get(1)
    mMenuItem=acontrol.get(2)
    'The physical control
    theControl = totalControl.get(1)
    'The control Index
    theCindex=totalControl.get(2)
    'Find the DocGUI
    theControlDoc=av.getproject.findGUI(aControl.get(0))
    if (theControlDoc=NIL) then
        MsgBox.Warning("The GUI "+aControl.get(0)+" cannot be found in the current project.","Script Eror")
        return(nil)
    end
    theMbar=av.getproject.findGUI(mDoc).GetMenuBar
    themenu=theMbar.findbylabel(mMenu)
    if (themenu=NiL) then
        MsgBox.Warning("The menu named "+mMenu+" is not here.","Script Eror")
        'return(nil)
    else
        thething=themenu.getcontrols.find(thecontrol)
        if (thething<>NIL) then
            themenu.remove(thecontrol)
        end
        'msgbox.info(themenu.GetControls.count.asstring,"")
        if (themenu.GetControls.count<1) then
            theMbar.remove(themenu)
        end
    end
end
for each totalControl in theToolMenuList
    'Get the control list from the Ext
    acontrol=totalControl.get(0)
    'Get the physical Control
    theControl = totalControl.get(1)
    'Get the Controls Index
    theCindex=totalControl.get(2)
    'Find the DocGUI for the Control
    theControlDoc=av.getproject.findGUI(aControl.get(0))
    if (theControlDoc=NIL) then
        MsgBox.Warning("The GUI "+aControl.get(0)+" cannot be found in the current project.","Script Eror")
        return(nil)
    end
    'This sequence finds the appropriate control set
    thecontrolset=av.getproject.findGUI(aControl.get(0)).GetToolBar
    'See if the control is in the set , if so remove it
    if (theControlSet.GetControls.find(theControl)<>NIL) then
    theControlSet.remove(theControl)
    theControlSet.selectdefault
    end
end
'****
'And the scripts delete themselves
'****
av.getproject.setmodified(true)
```

5

```
' Name:  Project.Create2TextFiles
'
' Author: Rod Schatz
'         Earth and Atmospheric Sciences Department
'         University of Alberta, 1998
'
' Date:  November 6, 1998
'
' Title: Create two text files to establish the parameters for the LA program
'
' Topics: Make, Write, MultiInput MsgBox, GetWorkDir
'
' Description:   This Avenue Script is one in a series of scripts used to create the ARCLA extension.  It creates the
Input
'              Parameters File and Optimize Name text files required by the Location- Allocation program.  The user
is
'              asked to input the number of iterations, number of facilities they require in terms of the lowest number
and
'              highest.  In addition, the script is setup to ask the user for  the Census Variable they are using and this
will
'              in turn be used create the Text Files Name
'
' Requires:   The user must have a Working Directory established
'
' Self:
'
' Returns:  A text file with values entered in by the user and another text file that stores the variable name.
'
' Updated:  February 3, 1999
'****
'Find the working directory to avoid the text file being created in the last directory used
'by the user or the default $HOME directory
'****
theWorkingDirectory = av.GetProject.GetWorkDir
theWorkingDirectoryString = av.GetProject.GetWorkDir.GetFullName
'****
'Use this line to test what the working directory is set as
'****
'MsgBox.Info(theWorkingDirectory.AsString, "LOOK AT ME!")
'****
'If a working directory is not set then the script will end
'****
if (File.Exists(theWorkingDirectory).Not) then
    MsgBox.Error("Unable to find a working directory" ++theWorkingDirectoryString++ "does not exist." + NL +
"Program Cannot continue" + NL + "Press OK to EXIT", "Error")
    exit
end
'****
'Create the MultiList Box Field Titles
'****
theTitle = "Enter the values for the Location-Allocation Parameters File"
theInstructions = "The Working Directory is"++theWorkingDirectory.AsString
theIterations = "Number of runs"
theStartingFacility = "Starting Facility"
theFinalFacility = "Ending Facility"
theVariableName = "Enter in the Census Variable Title"
'****
'Setup the two strings required by the MultiInput Box
'****
theLabelList = {theIterations, theStartingFacility, theFinalFacility, theVariableName}
theDefaultValueList = {"1","1","10", "1000"}
'****
```

```
'Create the List Box
'****
theValuesList = MsgBox.MultiInput(theInstructions, theTitle, theLabelList, theDefaultValueList)
'****
'Test to see if the variables are being passed correctly with a MsgBox
'****
'MsgBox.ListAsString(theValuesList, "Here are your values", "LOOK AT ME!" )
'****
'Error checking to handle the event if the user selects cancel
'****
if (theValuesList.Count = 0) then
    MsgBox.Error("Unable to Create the Parameters Text File" + NL + "One Field Parameter contains no DATA" +
NL + "Program Cannot continue" + NL + "Press OK to EXIT", "Error")
    return nil
end
'****
'The following few lines of code return the values the user inputted from the MutliInput MsgBox
'****
theNumberIterations = theValuesList.Get(0)
theNumberStartingFacility = theValuesList.Get(1)
theNumberFinalFacility = theValuesList.Get(2)
theNumberVariableName = theValuesList.Get(3)
'****
'Error checking for the Number of Iterations
'****
if (nil = theNumberIterations) then
    MsgBox.Error("Unable to Create the Parameters Text File" + NL + "One Field Parameter contains no DATA" +
NL + "Program Cannot continue" + NL + "Press OK to EXIT", "Error")
    return nil
end
'****
'Error checking for the Starting Facility Number
'****
if (nil = theNumberStartingFacility) then
    MsgBox.Error("Unable to open Theme's Table" + NL + "One Field Parameter contains no DATA" + NL +
"Program Cannot continue" + NL + "Press OK to EXIT", "Error")
    exit
end
'****
'Error checking for the Final Facility Number
'****
if (nil = theNumberFinalFacility) then
    MsgBox.Error("Unable to open Theme's Table" + NL + "One Field Parameter contains no DATA" + NL +
"Program Cannot continue" + NL + "Press OK to EXIT", "Error")
    exit
end
'****
'Error checking for the Variable Name
'****
if (nil = theNumberVariableName) then
    MsgBox.Error("Unable to open Theme's Table" + NL +"One Field Parameter contains no DATA" + NL +
"Program Cannot continue" + NL + "Press OK to EXIT", "Error")
    exit
end
'****
'Set up the File Naming Convention
'****
theTextFileName = theNumberVariableName.AsString+"_Input_Parameters.txt"
'****
'Create the Text File with default naming convention
'****
theTextFile = TextFile.Make(theTextFileName.AsFileName, #FILE_PERM_WRITE)
```

```
theTextFile.Write(theNumberIterations.AsString++theNumberStartingFacility.AsString++theNumberFinalFacility.AsS
tring + NL, 8)
theTextFile2 = TextFile.Make("Optimize_Name.txt".AsFileName, #FILE_PERM_WRITE)
theTextFile2.Write(theNumberVariableName.AsString + NL, 40)
theTextFile.Close
theTextFile2.Close
```

```
' Name:   Project.ImportFacilitiesSolutionFile
'
' Author: Rod Schatz
'         Earth and Atmospheric Sciences Department
'         University of Alberta, 1998
'
' Date:   December 14th, 1998
'
' Title:  Import Location-Allocation Output Facilities Solution File
'
' Topics: FileDialog.Show, Wild Card Search, ASCII Text,
'
' Description:  This Avenue Script is one in a series of scripts used to create the ARCLA extension.  This script  will
search
'               the current (working) directory for all files with the following naming convention
*_facility_locations.txt
'               (where the * denotes a variable name).  The script imports the text file, then asks the user if they would
like
'               to join the facility text file to an existing Census Canada Shapefile.  The script searches for the EAUID
'               field, if it exists it creates another field called EAuID_Number and copies all the EAuID values into it
the
'               new field as a numerical value instead of as a string.  After this operation is performed, it joins the two
'               tables with the Facility text file as the destination table.  Finally,  the script asks the user if they would
like
'               to add this file as an event theme to a chosen          View.  The last component will only work if the
Census
'               Shapefile is a point Theme.
'
' Requires:
'
' Self:
'
' Returns:
'
' Updated:  January 24th, 1999
'****
'Set up the lists required for the ReturnFile Dialog
'****
theSearchPatterns = {"*_facility_locations.txt"}
theTextLabel = {"*_facility_locations.txt (*.txt)"}
'****
'Find all the ASCII Text files with named *_Facility_Locations.txt
'where the * = some variable name
'****
theFacilityFile = FileDialog.ReturnFiles(theSearchPatterns, theTextLabel, "Add Facility Locations Table", 0)
'****
'Error Handling
'****
if (theFacilityFile.Count = 0) then exit end
'****
'Add the Text File to the Project (below is the default ESRI script)
'****
for each f in theFacilityFile
   v = VTab.Make(f, FALSE, FALSE)
   if (v.HasError) then
      if (v.HasLockError) then
         MsgBox.Error("Unable to acquire Read Lock for file " + f.GetBaseName, "")
      else
         MsgBox.Error("The file '" + f.GetBaseName + "' is not valid.", "")
      end
   else
      gui = av.GetProject.GetSelectedGUI
```

```
            if (gui.GetType <> "Table") then
                GUIName = "Table"
            else
                GUIName = gui.GetName
            end
        t = Table.MakeWithGUI(v, GUIName)
        t.SetName(v.GetName)
        t.GetWin.Open
        end
end
'****
'Ask the user if they would like to join the imported Table to an existing 'Shapefile.  If 'they respond NO the script will
'terminate.
'****
theUserChoice = MsgBox.YesNo("Would you like to join the Facility File to" ++ NL ++ "an existing Census
shapefile?", "Join File to Census Shapefile", True)
if (theUserChoice = false) then
    MsgBox.Warning("Program Cannot continue" + NL + "Press OK to EXIT", "Warning")
    exit
end
'****
'Now the Script asks the user to choose a View, and a Theme to join to the imported Text file to.
'****
theProject = av.GetProject
'****
'Create a list of the Views
'****
theViewList = List.Make
for each i in theProject.GetDocs
    if (i.Is(View)) then
        theViewList.Add(i)
    end
end
'****
'Ask the user to Select their View
'****
if (theViewList.Count = 0) then
    MsgBox.Warning("There are no VIEWS in your PROJECT: script will end", "Warning")
    return(nil)
end
theView = MsgBox.ListAsString(theViewList, "Please, select a View" + NL + "that contains your Census Shapefile",
"Select a VIEW")
if (theView = nil) then
    MsgBox.Warning("No VIEW selected: script will end", "Warning")
    return(nil)
end
'****
'Find the selected View's Themes
'****
theThemeList = theView.GetThemes
if (theThemeList = 0) then
    MsgBox.Warning("There are no THEMES in your VIEW: script will end", "Warning")
    return(nil)
end
'****
'Ask the user to select a Theme
'****
theTheme = MsgBox.ListAsString(theThemeList, "Select a THEME" + NL + "that you would like to join the Facility
File to:", "Select THEME")
if (theTheme = nil) then
    MsgBox.Warning("No THEME selected: script will end", "Warning")
    return(nil)
```

```
end
'****
'Get the Theme's Feature Table
'****
theFTab = theTheme.GetFTab
'****
'Search for the EAuID field.  This field is required as it needs 'to be modified to allow a join to take place between the
two
'tables
'****
theEAuIDField = theFTab.FindField("Eauid")
'****
'Error Checking to ensure EAuID field is present in the selected Table
'****
if (theEAuIDField = nil) then
    MsgBox.Warning("Selected Theme does not have the required EAuID field," + NL + "therefore a join cannot be
performed", + NL + "Program quitting", "Warning")
    exit
end
if (theFTab.FindField("Eauid_number")=nil) then
    theFTab.SetEditable(true)
    '****
    'Create the Number version of Eauid to facilitate the join
    '****
    theNewEAuIDField = Field.Make("EAuID_Number", #FIELD_LONG, 10, 0)
    theFTab.AddFields({theNewEAuIDField})
    theExpression = "([EAUID].AsNumber)"
    theFTab.Calculate(theExpression, theNewEAuIDField)
    theFTab.SetEditable(false)
else
end
'****
'Require the user to select the Name of the Facility file for the join.
'****
theDocList = theProject.GetDocs
'****
'Create a dictionary to store the data from the previous command
'****
theDict = Dictionary.Make(theDocList.Count)
for each i in theDocList
    if (i.is(Table)) then
        theDict.Add(i.GetName,i)
    end
end
theList = theDict.ReturnKeys
theTable = theDict.Get(MsgBox.ListAsString(theList, "Select the Name of your Imported Facility File", "Select
Imported Table"))
'****
'Error Handling - Warning Box if user does NOT select a Theme the script will Quit
'****
if (theTable = nil) then
    MsgBox.Error("No Table selected." + NL + "Program cannot continue" + NL + "Press OK to EXIT", "Error")
    return nil
end
'****
'Join the Facility text file to the chosen Theme using the EAuID_Number field.  This establishes a One to many join
with
'the imported table as the destination table.  Next, find the table and join field for the first table.
'****
VTab1 = theTheme.GetFTab
theJoinField1 = VTab1.FindField("EAuID_Number")
'****
```

```
'Error Checking to ensure EAuID_Number field is present in the selected Table
'****
if (theJoinField1 = nil) then
    MsgBox.Warning("Selected Theme does not have the required EAuID_Number field," + NL + "therefore a join
cannot be performed", + NL + "Program quitting", "Warning")
    exit
end
'****
' Find the table and join field for the second table.
'****
VTab2 = theTable.GetVtab
theJoinField2 = VTab2.FindField("Facility")
'****
'Error Checking to ensure Facility field is present in the selected Table
'****
if (theJoinField2 = nil) then
    MsgBox.Warning("Selected Facility File does not have the required Facility field," + NL + "therefore a join cannot
be performed", + NL + "Program quitting", "Warning")
    exit
end
'****
' Now perform the join.
'****
VTab2.Join(theJoinField2, VTab1, theJoinField1)
'****
'Ask the user if they would like to add the joined files as an event Theme.  If they
'respond no the script will end.
'****
theUserChoice2 = MsgBox.YesNo("Would you like add the Facility File to" ++ NL ++  "a View as a Event
Theme?"++ NL ++ "Theme MUST be a POINT Theme","Convert Facility File to Event Theme", True)
if (theUserChoice2 = false) then
    exit
end
'****
'Ask the user to Select their View to place the Event Theme into
'****
if (theViewList.Count = 0) then
    MsgBox.Warning("There are no VIEWS in your PROJECT: script will end", "Warning")
    return(nil)
end
theView = MsgBox.List(theViewList, "Please, select a View to insert the Facility File into:", "Select a VIEW")
if (theView = nil) then
    MsgBox.Warning("No VIEW selected: script will end", "Warning")
    return(nil)
end
'****
'Add Event Theme script Follows
'****
theVTab = theTable.GetVTab
theXField = theVTab.FindField("ealong")
'****
'Error Handling if Field is not present
'****
if (theXField = nil) then
    MsgBox.Error("Program requires a shapefile with two fields - EAlong and EAlat," + NL + "in order to create an
Event Theme", + NL + "Program quitting", "Error")
    exit
end
theYField = theVTab.FindField("ealat")
'****
'Error Handling if Field is not present
'****
```

```
if (theYField = nil) then
    MsgBox.Error("Program requires a shapefile with two fields - EAlong and EAlat," + NL + "in order to create an
Event Theme", + NL + "Program quitting", "Error")
    exit
end
theEventTheme = XYName.Make(theVTab, theXField, theYField )
theEventTheme = Theme.Make(theEventTheme)
theView.AddTheme(theEventTheme)
```

```
' Name:  Project.ImportFinalDemandsFile
'
' Author: Rod Schatz
'          Earth and Atmospheric Sciences Department
'          University of Alberta, 1998
'
' Date:   January 24th, 1999
'
' Title:  Import Location-Allocation Output Final Demands File
'
' Topics: FileDialog.Show, Wild Card Search, ASCII Text,
'
' Description:   This Avenue Script is one in a series of scripts used to create the ARCLA extension. This script  will
search
'                the current (working) directory for all files with the following naming convention
*_demand_allocations.txt
'                (where the * denotes a variable name). The script imports the text file, then asks the user if they would
'                like to join the facility text file to an existing Census Canada Shapefile.  The script searches for the
EAUID
'                field, if it exists it creates another field called EAuID_Number and copies all the EAuID values into it
the
'                new field as a numerical value instead of as a string.  After this operation is performed it joins the two
tables
'                with the Facility text file as the destination table.  Finally,  the script asks the user if they would like to
add
'                this file as an event theme to a chosen View.  The last component will only work if the Census
Shapefile is a
'                point Theme.
'
' Requires:
'
' Self:
'
' Returns:
'
' Updated:
'****
'Set up the lists required for the ReturnFile Dialog Box
'****
theSearchPatterns = {"*_Demand_Allocations.txt"}
theTextLabel = {"*_Demand_Allocations.txt (*.txt)"}
'****
'Find all the ASCII Text files with named *_Demand_Allocations.txt where the * = some variable name
'****
theDesireLinesFile = FileDialog.ReturnFiles(theSearchPatterns, theTextLabel, "Add Final Demands Table", 0)
'****
'Error Handling
'****
if (theDesireLinesFile.Count = 0) then exit end
'****
'Add the Text File to the Project (below is the default ESRI script)
'****
for each f in theDesireLinesFile
    v = VTab.Make(f, FALSE, FALSE)
    if (v.HasError) then
        if (v.HasLockError) then
            MsgBox.Error("Unable to acquire Read Lock for file " + f.GetBaseName, "")
        else
            MsgBox.Error("The file '" + f.GetBaseName + "' is not valid.", "")
        end
    else
        gui = av.GetProject.GetSelectedGUI
```

```
            if (gui.GetType <> "Table") then
                GUIName = "Table"
            else
                GUIName = gui.GetName
            end
        t = Table.MakeWithGUI(v, GUIName)
        t.SetName(v.GetName)
        t.GetWin.Open
        end
end
'****
'Ask the user if they would like to join the imported Table to an existing Shapefile.  If they respond NO the script will
'terminate.
'****
theUserChoice = MsgBox.YesNo("Would you like to join the Final Demands File to" ++ NL ++ "an existing Census
shapefile?", "Join File to Census Shapefile", True)
if (theUserChoice = false) then
    MsgBox.Warning("Program Cannot continue" + NL + "Press OK to EXIT", "Warning")
    exit
end
'****
'Now the Script to ask the user to choose a View, and a Theme to join to.
'****
theProject = av.GetProject
'****
'Create a list of the views
'****
theViewList = List.Make
for each i in theProject.GetDocs
    if (i.Is(View)) then
        theViewList.Add(1)
    end
end
'****
'Ask the user to Select their View
'****
if (theViewList.Count = 0) then
    MsgBox.Warning("There are no VIEWS in your PROJECT: script will end", "Warning")
    return(nil)
end
theView = MsgBox.ListAsString(theViewList, "Please, select a View" + NL + "that contains your Census Shapefile.",
"Select a VIEW")
if (theView = nil) then
    MsgBox.Warning("No VIEW selected: script will end", "Warning")
    return(nil)
end
'****
'Find the selected View's Themes
'****
theThemeList = theView.GetThemes
if (theThemeList = 0) then
    MsgBox.Warning("There are no THEMES in your VIEW: script will end", "Warning")
    return(nil)
end
'****
'Ask the user to select a Theme
'****
theTheme = MsgBox.ListAsString(theThemeList, "Select a THEME" + NL + "that you would like to join the
Demand_Allocations File to:", "Select THEME")
if (theTheme = nil) then
    MsgBox.Warning("No THEME selected: script will end", "Warning")
    return(nil)
```

```
end
'****
'Get the Theme's Feature Table
'****
theFTab = theTheme.GetFTab
'****
'Search for the EAuID field.  This field is required as it needs to be modified to allow a join to take place between the
two 'tables
'****
theEAuIDField = theFTab.FindField("Eauid")
'****
'Error Checking to ensure EAuID field is present in the selected Table
'****
if (theEAuIDField = nil) then
    MsgBox.Warning("Selected Theme does not have the required EAuID field," + NL + "therefore a join cannot be
performed", + NL + "Program quitting", "Warning")
    exit
end
if (theFTab.FindField("Eauid_number")=nil) then
    theFTab.SetEditable(true)
    '****
    'Create the Number version of Eauid to facilitate the join
    '****
    theNewEAuIDField = Field.Make("EAuID_Number", #FIELD_LONG, 10, 0)
    theFTab.AddFields({theNewEAuIDField})
    theExpression = "([EAUID].AsNumber)"
    theFTab.Calculate(theExpression, theNewEAuIDField)
    theFTab.SetEditable(false)
else
end
'****
'Require the user to select the Name of the Facility file for the join
'****
theDocList = theProject.GetDocs
'****
'Create a dictionary to store the data from the previous command
'****
theDict = Dictionary.Make(theDocList.Count)
for each i in theDocList
    if (i.is(Table)) then
        theDict.Add(i.GetName,i)
    end
end
theList = theDict.ReturnKeys
theTable = theDict.Get(MsgBox.ListAsString(theList, "Select the Name of your Imported Final Demands File", "Select
Imported Table"))
'****
'Error Handling - Warning Box if user does NOT select a Theme the script will Quit
'****
if (theTable = nil) then
    MsgBox.Error("No Table selected." + NL + "Program cannot continue" + NL + "Press OK to EXIT", "Error")
    return nil
end
'****
'Join Facility file to the chosen Theme using the EAuID_Number field.  The join will create a One to many join with
the
'imported table as the destination table Next, find the table and join field for the first table.
'****
VTab1 = theTheme.GetFTab
theJoinField1 = VTab1.FindField("EAuID_Number")
'****
'Error Checking to ensure EAuID_Number field is present in the selected Table
```

```
'****
if (theJoinField1 = nil) then
    MsgBox.Warning("Selected Theme does not have the required EAuID_Number field," + NL + "therefore a join
cannot be performed", + NL + "Program quitting", "Warning")
    exit
end
'****
'Find the table and join field for the second table.
'****
VTab2 = theTable.GetVtab
theJoinField2 = VTab2.FindField("Demand")
'****
'Error Checking to ensure Facility field is present in the selected Table
'****
if (theJoinField2 = nil) then
    MsgBox.Warning("Selected Facility File does not have the required Facility field," + NL + "therefore a join cannot
be performed", + NL + "Program quitting", "Warning")
    exit
end
'****
' Now perform the join.
'****
VTab2.Join(theJoinField2, VTab1, theJoinField1)
'****
'Ask the user if they would like to join this imported Table to an existing Shapefile.  If they respond no the script will
end.
'****
theUserChoice2 = MsgBox.YesNo("Would you like add the Facility File to" ++ NL ++ "a View as a Event Theme?",
"Convert Facility File to Event Theme", True)
if (theUserChoice2 = false) then
    exit
end
'****
'Ask the user to Select their View
'****
if (theViewList.Count = 0) then
    MsgBox.Warning("There are no VIEWS in your PROJECT: script will end", "Warning")
    return(nil)
end
theView = MsgBox.ListAsString(theViewList,  "Please, select a View to insert the Facility File into:", "Select a
VIEW")
if (theView = nil) then
    MsgBox.Warning("No VIEW selected: script will end", "Warning")
    return(nil)
end
'****
'Add Event Theme script Follows
'****
theVTab = theTable.GetVTab
theXField = theVTab.FindField("ealong")
'****
'Error Handling if Field is not present
'****
if (theXField = nil) then
    MsgBox.Error("Program requires a shapefile with two fields - EAlong and EAlat," + NL + "in order to create an
Event Theme", + NL + "Program quitting", "Error")
    exit
end
theYField = theVTab.FindField("ealat")
'****
'Error Handling if Field is not present
'****
```

```
if (theYField = nil) then
     MsgBox.Error("Program requires a shapefile with two fields - EAlong and EAlat," + NL + "in order to create an
Event Theme", + NL + "Program quitting", "Error")
     exit
end
theEventTheme = XYName.Make(theVTab, theXField, theYField )
theEventTheme = Theme.Make(theEventTheme)
theView.AddTheme(theEventTheme)
```

```
' Name: Project.SearchForLATextFiles
'
' Author: Rod Schatz
'          Earth and Atmospheric Sciences Department
'          University of Alberta, 1999
'
' Date:   January 19th, 1999
'
' Title: File Exists in Directory Path
'
' Topics:  File, WokingDirectory, FileName, MsgBox.YesNo
'
' Description:  This Avenue Script is one in a series of scripts used to create the ARCLA extension. This script checks
for the
'              existence of a the 4 text files (*_potential_sites.txt, * _demand_points.txt, * _input_parameters.txt,
'              optimize_name.txt, where the * character denotes a variable name) required by the Location-Allocation.
'              The script will search the working directory (set by the user using the File - Set Working Directory
'              function).  One problem might arise if the user accessed a directory other than the working directory to
add
'              a Theme or database Table as ArcView will read this directory instead of the working directory even
though
'              the working directory is set.  To resolve this click to add a Theme and navigate to the working directory
'              and select cancel.
'
' Requires:
'
' Self:
'
' Returns: Boolean answer (Yes it exists/No it doesn't)
'
' Updated:
'****
'Find the working directory to avoid the text file being created in the last directory used by the user or the $HOME
'directory
'****
theWorkingDirectory = av.GetProject.GetWorkDir
theWorkingDirectoryString = av.GetProject.GetWorkDir.GetFullName
theDirectoryAnswer =
MsgBox.YesNo("The program will search this Directory Path" + NL ++theWorkingDirectoryString++ NL + NL +
"Select YES if you would you like to continue?" + NL + "Select NO to manually change the current Working
Directory", "Conitnue?", TRUE)
'****
'Error Handling if the user selects NO
'****
if (theDirectoryAnswer.Not) then exit end
'****
'If a working directory is not set then the script will end
'****
if (File.Exists(theWorkingDirectory).Not) then
    MsgBox.Error("Unable to find a working directory" + NL + "Please SET you WORKING DIRECTORY and run
the program again." + NL +"Program Cannot continue" + NL + "Press OK to EXIT", "Error")
    exit
end
'****
'Create the Input Box (default value = 1000)
'****
theUserInput = MsgBox.Input("Enter in the variable name used for the three files:", "Input Variable Name", "1000")
'****
'Error Handling if the MsgBox is blank
'****
if (theUserInput = nil) then
```

```
    MsgBox.Error("Unable to find Files" + NL + "No Variable Name was entered" + NL + "Program Cannot continue"
+ NL + "Press OK to EXIT", "Error")
    return nil
end
'****
'find input parameters file
'****
TextFile1 =  theUserInput + "_input_parameters.txt".asstring
if (File.Exists(TextFile1.AsFileName).Not) then
    MsgBox.Error("File:" ++ TextFile1 ++ NL +"Does Not Exist: Exiting","Error")
    exit
end
TextFile1 = TextFile1.asFilename
'****
'find demand points file
'****
TextFile2 = theUserInput + "_demand_points.txt".AsString
if (File.Exists(TextFile2.AsFileName).Not) then
    MsgBox.Error("File:" ++ TextFile2 ++ NL +"Does Not Exist: Exiting","Error")
    exit
end
TextFile2 = TextFile2.asFilename
'****
'Find potential sites file
'****
TextFile3 = theUserInput + "_potential_sites.txt".AsString
if (File.Exists(TextFile3.AsFileName).Not) then
    MsgBox.Error("File:" ++ TextFile3 ++ NL +"Does Not Exist: Exiting","Error")
    exit
end
TextFile3 = TextFile3.asFilename
'****
'find optimize_name.txt
'****
TextFile4 =  "optimize_name.txt".asstring
if (File.Exists(TextFile4.AsFileName).Not) then
    MsgBox.Error("File:" ++ TextFile1 ++ NL +"Does Not Exist: Exiting","Error")
    exit
end
TextFile4 = TextFile4.asFilename
'****
'Return the finds to the user
'****
theReport = MsgBox.Info("Four files were located with the following names:"
            ++ NL ++ TextFile1.GetFullName
            ++ NL ++ TextFile2.GetFullName
            ++ NL ++ TextFile3.GetFullName
            ++ NL ++ TextFile4.GetFullName,
            "Parameters for LA Program are satisfied" )
```

```
' Name: Query.Census1996Data
'
' Author: Rod Schatz
'          Earth and Atmospheric Sciences Department
'          University of Alberta, 1998
'
' Date:   January 15th, 1999
'
' Title:  Open SQL Connect Dialog Box
'
' Topics:  SQL Con
'
' Description:  This Avenue Script is one in a series of scripts used to create the ARCLA extension.  The script opens
the
'               SQL Connect window to allow the user to query an external attribute database from within ArcView.
The
'               script calls the default SQL Window dialog box.  The script was crealed so that this functionality could
be
'               placed in the View menu.
'
' Requires:
'
' Self:
'
' Returns:
'
' Updated:

SQLWin.The.Open
```

```
' Name: Table.ConvertCensus2EventTheme
'
' Author: Rod Schatz
'           Earth and Atmospheric Sciences Department
'           University of Alberta, 1999
'
' Date:  February 4th, 1999
'
' Title:  Convert Census Database Table to an Event Theme
'
' Topics:
'
' Description:   This Avenue Script is one in a series of scripts used to create the ARCLA xtension.  This script will
convert a
'                  table with EALat (Latitude) and EALong (Longitude) to an Event Theme.  The table will then become
a
'                  point Theme in ArcView.
'
' Requires:  Two fields must be present in the selected Table EAlat and EALong (or x, y with the proper field names).
'            These two fields correspond to the latitude and longitude for each EA Centroid.  The required fields must
be
'            called EALat and EALong as the script looks for two fields with those names respectfully.
'
' Self:
'
' Returns:   A new Theme in a View
'
' Updated:
'****
' Get the ArcView Project Document
'****
theProject = av.GetProject
'****
' Select the Table Document
'****
theDocList = theProject.GetDocs
'****
'Create a dictionary to store the data from the previous command
'****
theDict = Dictionary.Make(theDocList.Count)
for each i in theDocList
     if (i.is(Table)) then
          theDict.Add(i.GetName,i)
     end
end
theList = theDict.ReturnKeys
theTable = theDict.Get(MsgBox.ListAsString(theList, "Choose a Table to display fields", "Select Census Table"))
'****
' Error Handling - Warning Box if user does NOT select a theme and script will Quit
'****
if (theTable = nil) then
     MsgBox.Error("No Table selected." + NL + "Program Cannot continue" + NL + "Press OK to EXIT", "Warning")
     return nil
end
'****
'Get the Table's VTab
'****
theVTab = theTable.GetVTab
'****
'The next section turns all the "active" fields off, which will allow the script to turn only the selected fields
'on in the section immediately after this one.
'****
```

```
theFields = theVTab.GetFields
'****
'Create a dictionary to store the data from the previous command
'****
theDict = Dictionary.Make(theFields.Count)
for each i in theFields
    if (i.is(Field)) then
        theDict.Add(i.GetName,i)
    end
end
'****
'Now the Script to ask the user to choose a View, and a Theme to join to.
'****
'theProject = av.GetProject
'****
'Create a list of the views
'****
theViewList = List.Make
for each i in theProject.GetDocs
    if (i.Is(View)) then
        theViewList.Add(i)
    end
end
'****
'Ask the user to Select their View
'****
if (theViewList.Count = 0) then
    MsgBox.Warning("There are no VIEWS in your PROJECT: script will end", "Warning")
    return(nil)
end
theView = MsgBox.ListAsString(theViewList, "Please, select a View" + NL + "to insert your Event Theme","Select a
VIEW")
if (theView = nil) then
    MsgBox.Warning("No VIEW selected: script will end", "Warning")
    return(nil)
end
'****
'Add Event Theme script Follows
'****
theVTab = theTable.GetVTab
theXField = theVTab.FindField("ealong")
'****
'Error Handling if Field is not present
'****
if (theXField = nil) then
    MsgBox.Error("Program requires a shapefile with two fields - EAlong and EAlat," + NL + "in order to create an
Event Theme", + NL + "Program quitting", "Error")
    exit
end
theYField = theVTab.FindField("ealat")
'****
'Error Handling if Field is not present
'****
if (theYField = nil) then
    MsgBox.Error("Program requires a shapefile with two fields - EAlong and EAlat," + NL + "in order to create an
Event Theme", + NL + "Program quitting", "Error")
    exit
end
theEventTheme = XYName.Make(theVTab, theXField, theYField )
theEventTheme = Theme.Make(theEventTheme)
theView.AddTheme(theEventTheme)
```

```
' Name: Table.DisplaySelectedFields
'
' Author: Rod Schatz
'           Earth and Atmospheric Sciences Department
'           University of Alberta, 1998
'
' Date:   March 10th, 1998
'
' Title: Export a Table by selecting the fields
'
' Topics:  Table, Fields, Export, VTab
'
' Description:   This Avenue Script is one in a series of scripts used to create the A R C L A extension.  This script
allows the
'                 user to select the fields they want to turn on in that table.  It will display a List Box that is filled with the
'                 Column titles, the user is asked to select the columns they wish to turn on.  The script will turn off all
'                 columns except for the user's choice.
'
' Requires:  A table in the project
'
' Self:
'
' Returns:
'
' Updated: February 3rd, 1999
'***
'Variable initialization
'****
t = "Table Display selected Fields"
'****
' Get the ArcView Project Document
'****
theProject = av.GetProject
'****
' Select the Table Document
'****
theDocList = theProject.GetDocs
'****
'Create a dictionary to store the data from the previous command
'****
theDict = Dictionary.Make(theDocList.Count)
for each i in theDocList
    if (i.is(Table)) then
        theDict.Add(i.GetName,i)
    end
end
theList = theDict.ReturnKeys
theTable = theDict.Get(MsgBox.ListAsString(theList, "Choose a Table to display fields", t))
'****
' Error Handling - Warning Box if user does NOT select a theme and script will Quit
'****
if (theTable = nil) then
    MsgBox.Error("No table selected." + NL + "Program Cannot continue" + NL + "Press OK to EXIT", "Warning")
    return nil
end
'****
'Get the Table's VTab
'****
theVTab = theTable.GetVTab
'****
'The next section turns all the "active" fields off which will allow 'the script to turn only the selected fields
'on in the section immediately after this one
```

```
'****
theFields = theVTab.GetFields
'****
'Create a dictionary to store the data from the previous command
'****
theDict = Dictionary.Make(theFields.Count)
for each i in theFields
    if (i.is(Field)) then
        theDict.Add(i.GetName,i)
    end
end
'****
'Turns all the fields off that were identified in theDict
'****
for each theFields in theDict
    theFields.SetVisible(false)
end
'****
'Display the fields available in the user selected table
'****
theFieldList = MsgBox.MultiList(theVTab.GetFields, "Select Field(s) (use the shift key for multiple selections)",
"Field(s) Selection")
'****
'Error Handling -  Warning  message if the user does not select any fields the script will close
'****
if (theFieldList = nil) then
    MsgBox.Warning("No fields selected, program closing", "Warning")
    return (nil)
end
'****
'Now take the user chosen fields and set them active
'****
theField = theVTab.FindField(theFieldList.AsString)
for each theField in theFieldList
    theField.SetVisible(true)
end
'****
'Open the selected table to show the user that they have chosen the correct fields and also to prepare the script to work
with
'the export script
'****
if (theTable.GetWin.IsOpen) then
    theTable.GetWin.Close
else
    exit
end
theTable.GetWin.Open
```

```
' Name:     Table.ExportDemandPoints
'
' Author:   Rod Schatz
'           University of Alberta
'           Edmonton, Alberta, 1998
'
' Date:  June 12th, 1998
'
' Title:    Export Demand Points for use with an external L.A. program.
'
' Topics:  FileName, Export, SetVisible, Ftab, Dictionary, Shuffle, GetFields.
'
' Description:   This Avenue Script is one in a series of scripts used to create the A R C L A extension.  The purpose of
this
'                script is to automate the process of exporting a text file to be used as a demand points file for use with
the
'                location-allocation program.  This script is designed to be run from a button placed on ArcView's GUI
or
'                from a pull down menu.  The script will find              all the Views in the open Project then prompt the
user to
'                select a Theme from the chosen View.  The Theme chosen should contain the Shapefile that the user
wishes
'                to use as their                demand nodes Shapefile.  The script will automatically turn on the necessary
fields except for
'                one.  The user must choose this field.  This particular field will be used as the demand weight in the
'                location-allocation program.  This field will also be used to name the output file.  The chosen Table will
be
'                exported as a Text (ASCII format) file with 4 fields (Eauid, Ealat, Ealong, and the demand weight
chosen
'                by the user).  The script will also re-order the fields to ensure that they are located in the proper order
'                Eauid, Ealat, Ealong and a Demand Weight.  The location-allocation program requires the fields are
'                located in the proper order.
'
' Requires:   A Point Shapefile created from the 1996 Census Canada data set which must contain these three fields -
Eauid
'             (Enumeration Area Identifier), Ealat (EA latitude), Ealong (EA longitude) plus one more field to be used as
a
'             demand weight in an external location allocation  program.
'
' Self:
'
' Returns:
'
' Updated:  July 9th, 1998
'****
'Start by finding all the Project's Views
'****
theProject = av.GetProject
'****
'Create a list of the Views in the Project
'****
theViewList = List.Make
for each i in theProject.GetDocs
    if (i.Is(View)) then
        theViewList.Add(I)
    end
end
'****
'Ask the user to select the View containing their Demand Nodes Theme
'****
if (theViewList.Count = 0) then
```

```
    MsgBox.Error("This Project does not contain any Views" + NL + "Program Cannot continue" + NL + "Press OK to
EXIT", "Error")
    exit
end
theView = MsgBox.List(theViewList, "Please, select the View that" + NL + "contains your Demand Nodes
(POINT)Theme", "Select a VIEW")
if (theView = nil) then
    MsgBox.Error("No VIEW selected: script will end" + NL + "Program Cannot continue" + NL + "Press OK to
EXIT", "Error")
    exit
end
'****
'Find the selected View's Themes
'****
theThemeList = theView.GetThemes
if (theThemeList = 0) then
    MsgBox.Error("This View does not contain any Themes" + NL + "Program Cannot continue" + NL + "Press OK to
EXIT", "Error")
    exit
end
'****
'Ask the user to select a Theme to be used as their Demand Nodes
'****
theTheme = MsgBox.List(theThemeList, "Select a THEME to be used as your DEMAND NODES:", "Select a
Demand Nodes Theme")
if (theTheme = nil) then
    MsgBox.Error("No THEME selected" + NL + "Program Cannot continue" + NL + "Press OK to EXIT", "Error")
    exit
end
'****
'Error Checking to Ensure the chosen Theme is a point Theme
'****
if ((theTheme.GetSrcName.GetSubName = "point").Not) then
    MsgBox.Error("The chosen Theme must be a point theme" + NL + "Program Cannot continue" + NL + "Press OK
to EXIT", "Error")
    return nil
end
'****
'Turn off all the fields in the selected Theme's Table.  This will allow the script to export only the following
'fields (EAuid, Ealat, Ealong, and a demand variable chosen by the user).
'****
theFTab = theTheme.GetFTab
theFields = theFTab.GetFields
theDict = Dictionary.Make(theFields.Count)
for each i in theFields
    if (i.is(Field)) then
        theDict.Add(i.GetName,i)
    end
end
'****
'Turn all the fields off that were identified in theDict (Dictionary)
'****
for each theField in theDict
    theField.SetVisible(false)
end
'****
'Turn on the default fields (EAuid, Ealat, Ealong), confirm that the table has the proper fields, check for the
'EAuid field and set it visible
'****
theIDField = theFTab.FindField("Eauid")
if (theIDField = nil) then
```

```
    MsgBox.Error("This Theme's table does not contain the necessary field called EAuid" + NL +"Program Cannot
continue" + NL + "Press OK to EXIT", "Error")
    exit
else
    theIDField.SetVisible(true)
end
'****
'Check for the EAlat field and set it visible
'****
theLatField = theFTab.FindField("Ealat")
if (theLatField = nil) then
    MsgBox.Error("This Theme's table does not contain the necessary field called EAlat" + NL + "Program Cannot
continue" + NL + "Press OK to EXIT", "Error")
    exit
else
    theLatField.SetVisible(true)
end
'****
'Check for the EAlong field and set it visible
'****
theLongField = theFTab.FindField("Ealong")
if (theLongField = nil) then
    MsgBox.Error("This Theme's table does not contain the necessary field called EAlong" + NL + "Program Cannot
continue" + NL + "Press OK to EXIT", "Error")
    exit
else
    theLongField.SetVisible(true)
end
'****
'Ask the user to choose a Field to be used as their Demand Nodes Weight
'****
theFieldList = MsgBox.List(theFTab.GetFields, "Select a Field to be used as your Demand Variable:", "Demand
Weight Selection")
'****
'Warning  message if the user does not select any fields the script will close
'****
if (theFieldList = nil) then
    MsgBox.Error("No field selected" + NL + "Program Cannot continue" + NL + "Press OK to EXIT", "Error")
    return (nil)
    exit
end
'****
'Now it is safe to obtain the FTab's Selection
'****
theBitmap = theFTab.GetSelection
'****
'Establish the Query Syntax (string) Statement
'****
theQuery = "([ "++ theFieldList. GetName ++" ] > 0.000000)"
'****
'Perform the Query and update selection
'****
theFTab.Query(theQuery, theBitmap, #VTAB_SELTYPE_AND)
theFtab.UpdateSelection
'****
'Now take the chosen Demand field and set it active
'****
theField = theFTab.FindField(theFieldList.AsString)
theField.SetVisible(true)
if (nil = VTab) then
    MsgBox.Error("Unable to open Theme's Table" + NL +"Program Cannot continue" + NL + "Press OK to EXIT",
"Error")
```

```
    exit
end
'****
'Find the "Eauid", "Ealat" and "Ealong" fields
'****
theFieldsFTab = theTheme.GetFTab
theEAuidField = theFieldsFTab.FindField("Eauid")
theEAlatField = theFieldsFTab.FindField("Ealat")
theEAlongField = theFieldsFTab.FindField("Ealong")
theFieldOrderList = theFieldsFTab.GetFields
'****
"Re-order the fields listed above
'****
theFieldOrderList.Shuffle(theEAuidField, 0)
theFieldOrderList.Shuffle(theEAlatField, 1)
theFieldOrderList.Shuffle(theEAlongField, 2)
theFieldsFTab.Refresh
'****
'Export the selected table as an ASCII file (*.txt)
'****
theNewVTab = theTheme.GetFTab
theNameFTab = theFTab.FindField(theFieldList.AsString)
if (nil = theNewVTab) then
    MsgBox.Error("Unable to open Theme's Table" + NL +"Program Cannot continue" + NL + "Press OK to EXIT",
"Error")
    exit
end
theNameFTab = theFTab.FindField(theFieldList.AsString)
'****
'Get the Field Name
'****
theNewName = theNameFTab.GetName+"_demand_points.txt"
'****
'Use the following line to ensure that it is naming the table correctly
'****
'MsgBox.Info(theNewName.AsString, "")
'****
'Get a file name
'****
aFileName = FileDialog.Put(theNewName.AsFileName, "*.txt", "Create Demand Nodes Table")
if (nil = aFileName) then
    MsgBox.Error("Unable to Save File" + NL + "Program Cannot continue" + NL + "Press OK to EXIT", "Error")
    exit
end
'****
'Export the table to a text file
'****
theNewVTab.Export(aFileName, DText, True)
```

```
' Name:    Table.ExportFacilityPoints
'
' Author:  Rod Schatz
'          University of Alberta
'          Edmonton, Alberta, 1998
'
' Date:  July 12th, 1998
'
' Title:   Export Facility Points for use with an external L.A. program.
'
' Topics:  FileName, Export, SetVisible, Ftab, Dictionary, Shuffle, GetFields.
'
' Description:   This Avenue Script is one in a series of scripts used to create the A R C L A extension. The purpose of
this
'                script is to automate the process of exporting a text file to be used as the potential facility locations file
with
'                the location allocation program.  This script is designed to be run from a button placed on ArcView's
GUI
'                or from a pull down menu.  The script will find all the Views in the open Project then prompt the user
to
'                select a Theme from the chosen View.  The Theme chosen should contain the Shapefile that the user
wishes
'                to use as their            Facility Nodes Shapefile.  The script will automatically turn on the necessary
fields. One
'                field will be used to name the output file (the same variable name used as the demand nodes file.  The
'                chosen Table will be exported as a Text (ASCII format) file with 3 fields (Eauid, Ealat, Ealong). The
script
'                will also re-order the fields to ensure that they are located in the proper order Eauid, Ealat and Ealong.
'
' Requires:   A Point Shapefile created from the 1996 Census Canada data set which must contain these three fields -
Eauid
'             (Enumeration Area Identifier), Ealat (EA latitude), Ealong (EA longitude) plus one more field to be used as
a
'             demand weight in an external location allocation  program.
'
' Self:
'
' Returns:
'
' Updated:
theProject = av.GetProject
'****
'Create a list of the Views in the Project
'****
theViewList = List.Make
for each i in theProject.GetDocs
    if (i.Is(View)) then
        theViewList.Add(i)
    end
end
'****
'Ask the user to select the View containing their Facility Nodes Theme
'****
if (theViewList.Count = 0) then
    MsgBox.Error("This Project does not contain any Views" + NL + "Program Cannot continue" + NL + "Press OK to
EXIT", "Error")
    end
theView = MsgBox.List(theViewList, "Please, select the View that" + NL + "contains your Facility Nodes (POINT)
Theme", "Select a VIEW")
if (theView = nil) then
    MsgBox.Error("No VIEW selected: script will end" + NL + "Program Cannot continue" + NL + "Press OK to
EXIT", "Error")
```

```
    exit
end
'****
'Find the selected View's Themes
'****
theThemeList = theView.GetThemes
'****
'Error Handling to ensure the user has a View open as the active document
'****
if (theThemeList = 0) then
    MsgBox.Error("This View does not contain any Themes" + NL + "Program Cannot continue" + NL + "Press OK to
EXIT", "Error")
    exit
end
'****
'Ask the user to select a Theme to be used as their facility nodes
'****
theTheme = MsgBox.List(theThemeList, "Select a THEME to be used as your FACILITY NODES:", "Select a Facility
Nodes Theme")
if (theTheme = nil) then
    MsgBox.Error("No THEME selected" + NL + "Program Cannot continue" + NL + "Press OK to EXIT", "Error")
end
'****
'Turn off all the fields in the selected Theme's Table.  This will allow the script to export EAuid, Ealat, Ealong.
'****
theFTab = theTheme.GetFTab
theFields = theFTab.GetFields
theDict = Dictionary.Make(theFields.Count)
for each i in theFields
    if (i.is(Field)) then
        theDict.Add(i.GetName,i)
    end
end
'****
'Turn all the fields off that were identified in theDict (Dictionary)
'****
for each theField in theDict
    theField.SetVisible(false)
end
'****
'Turn on the default fields (EAuid, Ealat, Ealong), and confirm that the table has the proper fields
'Check for the EAuid field and set it visible
'****
theIDField = theFTab.FindField("Eauid")
if (theIDField = nil) then
    MsgBox.Error("This Theme's table does not contain the necessary field called EAuid" + NL + "Program Cannot
continue" + NL +  "Press OK to EXIT", "Error")
    exit
else
    theIDField.SetVisible(true)
end
'****
'Check for the EAlat field and set it visible
'****
theLatField = theFTab.FindField("Ealat")
if (theLatField = nil) then
    MsgBox.Error("This Theme's table does not contain the necessary field called EAlat" + NL + "Program Cannot
continue" + NL +  "Press OK to EXIT", "Error")
    exit
else
    theLatField.SetVisible(true)
end
```

```
'****
'Check for the EAlong field and set it visible
'****
theLongField = theFTab.FindField("Ealong")
if (theLongField = nil) then
    MsgBox.Error("This Theme's table does not contain the necessary field called EAlong" + NL + "Program Cannot
continue" + NL + "Press OK to EXIT", "Error")
    exit
else
    theLongField.SetVisible(true)
end
'****
'Ask the user to choose a Field that corresponds to their facility nodes
'****
theFieldList = MsgBox.List(theFTab.GetFields, "Select a Field to be used as your Facility Nodes File:", "Facility
Nodes Selection")
'****
'Warning  message if the user does not select any fields the script will close
'****
if (theFieldList = nil) then
    MsgBox.Error("No field selected" + NL + "Program Cannot continue" + NL + "Press OK to EXIT", "Error")
    return (nil)
    exit
end
'****
'Now take the chosen demand variable name field and set it active
'****
'theField = theFTab.FindField(theFieldList.AsString)
'theField.SetVisible(true)
'****
'Find the "Eauid", "Ealat" and "Ealong" fields
'****
theFieldsFTab = theTheme.GetFTab
theEAuidField = theFieldsFTab.FindField("Eauid")
theEAlatField = theFieldsFTab.FindField("Ealat")
theEAlongField = theFieldsFTab.FindField("Ealong")
theFieldOrderList = theFieldsFTab.GetFields
'****
''Re-order the fields listed above
'****
theFieldOrderList.Shuffle(theEAuidField, 0)
theFieldOrderList.Shuffle(theEAlatField, 1)
theFieldOrderList.Shuffle(theEAlongField, 2)
theFieldsFTab.Refresh
'****
'Export the selected table as an ASCII file (*.txt)
'****
theNewVTab = theTheme.GetFTab
theNameFTab = theFTab.FindField(theFieldList.AsString)
if (nil = theNewVTab) then
    MsgBox.Error("Unable to open Theme's Table" + NL + "Program Cannot continue" + NL + "Press OK to EXIT",
"Error")
    exit
end
theNameFTab = theFTab.FindField(theFieldList.AsString)
'****
'Get the Field Name
'****
theNewName = theNameFTab.GetName+"_potential_sites.txt"
'****
'Use the following line to ensure that it is naming the table correctly
'****
```

```
'MsgBox.Info(theNewName.AsString, "")
'****
'Get a file name
'****
aFileName = FileDialog.Put(theNewName.AsFileName, "*.txt", "Create Facility Nodes Table")
if (nil = aFileName) then
    MsgBox.Error("Unable to Save File" + NL + "Program Cannot continue" + NL + "Press OK to EXIT", "Error")
    exit
end
'****
'Export the table to a dbase file
'****
theNewVTab.Export(aFileName, DText, True)
```

```
' Name:     Table.ExportOnlySelectedFields
'
' Author:   Rod Schatz
'           Earth and Atmospheric Sciences Department
'           University of Alberta, 1998
'
' Date:  March 6th, 1998
'
' Title:  Export a Table by selecting the Fields
'
' Topics:     Table, Fields, Export, VTab, Dictionary, MsgBox, FindField,
'       GetFields, Set Visible
'
' Description:   This Avenue Script is one in a series of scripts used to create the A R C L A extension.  This script
allows
'                the user to select a table they wish to export from ArcView as a text file and then prompts the user to
select
'                only the fields they wish to export.  The script will export only the fields selected.
'
' Requires:  A table in the project
'
' Self:
'
' Returns:
'
' Updated: June 10th, 1998
'****
'Variable initialization
'****
t = "Table Export Script"
'****
' Get the active Project
'****
theProject = av.GetProject
'****
' Find the available Tables
'****
theDocList = theProject.GetDocs
'****
'Create a dictionary to store the data from the previous command
'****
theDict = Dictionary.Make(theDocList.Count)
for each i in theDocList
    if (i.is(Table)) then
        theDict.Add(i.GetName,i)
    end
end
theList = theDict.ReturnKeys
theTable = theDict.Get(MsgBox.ListAsString(theList, "Choose a Table to export", t))
'****
' Warning Box if user does NOT select a Table and script will quit
'****
if (theTable = nil) then
    MsgBox.Error("No Table selected." + NL + "Program Cannot continue" + NL + "Press OK to EXIT", "Error")
    exit
end
'****
'Get the Table's VTab
'****
theVTab = theTable.GetVTab
'****
'The next section turns all the "active" Fields off.  This will allow the script to turn on only the selected
```

```
'Fields on in the section immediately after this one
'****
theFields = theVTab.GetFields
'****
'Create a dictionary to store the data from the previous command
'****
theDict = Dictionary.Make(theFields.Count)
for each i in theFields
    if (i.is(Field)) then
        theDict.Add(i.GetName,i)
    end
end
'****
'Turns all the Fields off that were identified in Dictionary (theDict)
'****
for each theFields in theDict
    theFields.SetVisible(false)
end
'****
'Display the Fields available in the table selected by the user
'****
theFieldList = MsgBox.MultiList(theVTab.GetFields, "Select Field(s) to export (use the shift key for multiple
selections)", "Field(s) Selection")
'****
'Warning message if the user does not select any Fields the script will close
'****
if (theFieldList = nil) then
    MsgBox.Error("No Fields selected." + NL + "Program Cannot continue" + NL + "Press OK to EXIT", "Error")
    return (nil)
end
'****
'Now take the chosen fields and set them active
'****
theField = theVTab.FindField(theFieldList.AsString)
for each theField in theFieldList
    theField.SetVisible(true)
end
'****
'Open the selected table to show the user the fields they have chosen.  This will allow the script to work
'with the export section
'****
theTable.GetWin.Open
'****
'Export script follows:
'The script continues to look for the active document (selected by the user from theTable request)
'****
theTable = av.GetActiveDoc
formats = {"dBASE", "INFO", "Delimited Text"}
format = MsgBox.ListAsString(formats, "Export Format:", "Export Table")
if (format = "dBASE") then
    theClass = DBASE
    theFilter = "*.dbf"
    theExt = "dbf"
elseif (format = "Delimited Text") then
    theClass = DTXT
    theFilter = "*.txt"
    theExt = "txt"
elseif (format = "INFO") then
    theClass = INFO
    theFilter = "arcdr9"
    theExt = ""
else
```

```
    return nil
end
theFileName=FileDialog.Put(av.GetProject.MakeFileName("table", theExt), theFilter,"Export Table")
if (theFileName = NIL) then return nil end
theVTab = theTable.GetVTab
ext     = theFileName.GetExtension
if (((theClass.IsSubclassOf(DTXT)) or (theClass.IsSubClassOf(DBASE))) and (ext <> theExt)) then
theFileName.SetExtension(theExt)
end
if (theVTab.GetSelection.Count=0) then
    theVTab.Export(theFileName, theClass, FALSE)
    av.ShowMsg("All records written to"++theFileName.GetBaseName)
else
    theVTab.Export(theFileName, theClass, TRUE)
    av.ShowMsg("Selected records written to"++theFileName.GetBaseName)
end
if ((System.GetOS = #SYSTEM_OS_MAC) and ((format = "INFO").not)) then
    Script.Make("MacClass.SetDocInfo(SELF, Table)").DoIt(theFileName)
end
'****
'Close the open Table window
'****
theTable.GetWin.Close
```

```
' Name: Table.FindDuplicatesValues
'
' Author:  Mark Cederholm
'          e-mail: piersen@primenet.com
'          http://www.primenet.com/~piersen/arcview/table.htm
'
'Modified By:  Rod Schatz
'              Earth and Atmospheric Sciences Department
'              University of Alberta, 1998
'
' Date:  December 30th, 1998
'
' Title:  Table Find Duplicate Field Values
'
' Topics:
'
' Description:   This Avenue Script is one in a series of scripts used to create the A R C L A extension. Script scans the
'                current table document and labels duplicate records based on a one or more fields (shape fields are
ignored).
'                Two options are available:
'                1. Update the selection set to include duplicates
'                2. Add a character field called "Match" (if it doesn't already exist) and set it to "True" for a duplicate
value
'                and "False " for no duplicates if records are selected, the selection set alone is analyzed.
'
' Requires:
'
' Self:
'
' Returns:  A table with a new field indicating whether or not duplicate values are present
'
' Updated:  February 8th, 1999
theTable = av.GetActiveDoc
'****
'Error Handling to ensure the user has a View open as the
'active document
'****
if (theTable.Is(Table).Not) then
    MsgBox.Warning("A Table must be active (open)" + NL + "Program Cannot continue" + NL + "Press OK to
EXIT", "Warning")
    return nil
end
theVTab = theTable.GetVTab
'****
'Generate field list for analysis
'****
theFieldList = List.Make
for each f in theVTab.GetFields
    theFieldName = f.GetName
    if ((theFieldName = "Shape") or (theFieldName = "Match")) then
        continue
    end
    theFieldList.Add(theFieldName)
end
'****
'Obtain a list fields to analyze
'****
theFieldAnalysis = "Select field(s) to analyze"
theTitle = "Find Duplicate Fields"
theResult = MsgBox.MultiListAsString(theFieldList,theFieldAnalysis,theTitle)
if (theResult.Count = 0) then
    Return nil
```

```
end
theFieldList= List.Make
for each theFieldName in theResult
    f = theVtab.FindField(theFieldName)
    theFieldList.Add(f)
end
'****
'Get option to update selection or calculate field
'****
c_list = {"Reselect duplicates","Calculate duplicate field"}
theResult = MsgBox.ListAsString(c_list,"Select option",theTitle)
if (theResult = nil) then
    return nil
end
option = c_list.FindByValue(theResult) + 1
if (option = 2) then
    '****
    'Begin Comparison
    '****
    if (theVTab.StartEditingWithRecovery.Not) then
        MsgBox.Error("Cannot edit selected Table","theTitle")
        return nil
    end
    theVTab.BeginTransaction
    df = theVTab.FindField("Match")
    if (df = nil) then
        '****
        'Add the "Match" Field
        '****
        df = Field.Make("Match",#FIELD_CHAR,8,0)
        theVTab.AddFields({df})
    end
end
'****
'Get set of records to analyze
'****
theRecs = theVTab.GetSelection
if (option = 1) then
    theBitmap = BitMap.Make(theRecs.GetSize)
end
NumRec = theRecs.Count
if (NumRec = 0) then
    theRecs = theVTab
    NumRec = theVTab.GetNumRecords
end
'****
'Analyze records
'****
key_list = List.Make
dc = 9.AsChar
av.ClearMsg
av.ClearStatus
av.ShowMsg("Analyzing records...")
Rec = 0
for each r in theRecs
    '****
    'Generate string representing key values delimited by the 9 (tab) character
    '****
    keystring = ""
    for each f in theFieldList
        w = f.GetWidth
        p = f.GetPrecision
        if (f.IsTypeString) then
            val_s = theVtab.ReturnValue(f,r)
            len = val_s.Count
                if (len < w) then
                val_s = val_s + String.MakeBuffer(w - len)
                end
        elseif (f.GetType = #FIELD_DATE) then
            val_d = theVtab.ReturnValue(f,r)
            if (val_d.IsNull) then
                val_s = String.MakeBuffer(w)
            else
                fs = "yyyyMMdd"
                val_d.SetFormat(fs)
                val_s = val_d.AsString
            end
        else
            val_n = theVtab.ReturnValue(f,r)
            if (val_n.IsNull.Not) then
                fs = String.MakeBuffer(w)
                fs = fs.Translate(" ","d")
                if (p > 0) then
                    fs = fs.Left(w - p - 1)+"."+fs.Right(p)
                end
                val_n = theVtab.ReturnValue(f,r)
                val_n.SetFormat(fs)
                val_s = val_n.AsString
            else
                val_s = String.MakeBuffer(w)
            end
        end
    keystring = keystring + val_s + dc
    end
    '****
    'Add the entry if not duplicate
    '****
    IsDup = false
    if (key_list.Count = 0) then
        key_list.Add(keystring)
    elseif (key_list.FindByValue(keystring) = -1) then
        key_list.Add(keystring)
    else
        IsDup = true
    end
    '****
    'Label record(s)
    '****
    if (option = 2) then
        if (IsDup) then
            theVTab.SetValue(df,r,"True")
    else
        theVTab.SetValue(df,r,"False")
        end
    else
        if (IsDup) then
            theBitmap.Set(r)
        end
    end
    Rec = Rec + 1
    av.SetStatus(100*Rec/NumRec)
end
key_list = nil
av.PurgeObjects
```

```
if (option = 2) then
    theVTab.EndTransaction
    theVTab.StopEditingWithRecovery(true)
else
    theVTab.SetSelection(theBitMap)
    theVTab.UpdateSelection
end
return nil
```

```
' Name:    Table.QueryDemandShapefile
'
' Author:   Rod Schatz
'           University of Alberta
'           Edmonton, Alberta, 1998
'
' Date:     July 12th, 1998
'
' Title:  Query Demand Nodes Solution
'
' Topics: MsgBox, List, Export, FTab, GetSelection, Update, Query
'
' Description:  This Avenue Script is one in a series of scripts used to create the A R C L A extension. The script will
query
'               a Theme based on a field labeled as "P".  This field corresponds to the output file for the location-
allocation
'               program.  The *_demand_allocations.txt file should have been joined to a an existing point Theme
using a
'               one-to-many join.  This script is designed to be run on the Theme containing the join.  It will
breakdown
'               the solutions from the location-allocation program based on the P values.  This is required if the user
wishes
'               to create or draw the desire lines.
'
'
' Requires:
'
' Self:
'
' Returns:
'
' Updated:  January 3rd, 1999
'****
'Variable Initialization
'****
theView = av.GetActiveDoc

theTheme = theView.GetActiveThemes.Get(0)

'****
'Error Checking to ensure the chosen Theme has an attribute table (FTheme)
'****
if (theTheme.Is(FTheme).Not) then
  MsgBox.Error("Active theme is not FTheme", "Error")
  return nil
end
'****
'Create a List Box with "P" values
'****
theNumberList = { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25 }
theUserChoice = MsgBox.ListAsString(theNumberList, "Please select a P value to extract from the database:", "Select
a P value " )
'****
'Error handling, script will end if the user does not select a P value
'****
if (theUserChoice = nil) then
    MsgBox.Warning("No selection made." + NL + "Program Cannot continue" + NL + "Press OK to EXIT",
"Warning")
    return nil
end
'****
'Get the Themes Feature Table
'****
```

```
theVTab = theTheme.GetFTab
'****
'Clear the highlighted selection, to ensure the user does not select multiple values of "P"
'****
theTheme.GetFTab.GetSelection.ClearAll
theTheme.GetFTab.UpdateSelection
theBitMap = theVTab.GetSelection
'****
'Find the Field labeled as "P"
'****
thePField = theVTab.FindField("P")
'****
'Error Handling, if the field "P" is not present the script will end
'****
if (thePField = nil) then
    MsgBox.Error("This TABLE is not in the proper format for this script: script will end" + NL + "Program Cannot
continue" + NL + "Press OK to EXIT", "Error")
    exit
end
'****
'Set up and execute the Query
'****
theQueryString = "([P] = "+ theUserChoice.AsString +")"
theVTab.Query(theQueryString, theBitMap, #VTAB_SELTYPE_NEW)
theVTab.SetSelection(theBitMap)
'****
'Convert the selection to a new shapefile
'****
'MsgBox.YesNo("Would you like to convert the selection to a new shapefile?", "Convert to Shapefile", TRUE)
theConvertedFile = av.GetActiveDoc
for each t in theView.GetActiveThemes
    if (t.Is( FTHEME ).Not) then
        if (t.CanExportToFtab.Not) then
            continue
        end
    end
    if (t.Is (FTHEME).Not) then
        def = av.GetProject.MakeFileName("theme", "shp")
        def = FileDialog.Put(def, "*.shp", "Convert " + t.getName)
        if (def = NIL) then
            return NIL
        end
        anFTab = t.ExportToFtab(def)
    else
        tbl = t.GetFTab
        attribVis = FALSE
        for each f in tbl.GetFields
            if ((f.IsVisible) and not (f.IsTypeShape)) then
                attribVis = TRUE
                break
            end
        end
    shapeVis = tbl.FindField("Shape").IsVisible
    if ((attribVis and shapeVis).Not) then
        continue
    end
    def = av.GetProject.MakeFileName("theme", "shp")
    def = FileDialog.Put(def, "*.shp", "Convert " + t.getName)
    if (def = NIL) then return nil end
    shpfld = (tbl.FindField("Shape"))
    if (shpfld.IsVisible.Not) then
        shpfld.SetVisible(shpfld.IsVisible.Not)
    WasNotVisible = TRUE
    else
    WasNotVisible = FALSE
    end
    anFTab = tbl.Export(def, Shape, tbl.GetSelection.Count > 0)
    if (anFTab.HasError) then
        if (anFTab.HasLockError) then
            MsgBox.Error("Unable to acquire Write Lock for file " + def.GetBaseName,"")
        else
            MsgBox.Error("Unable to create " + def.GetBaseName,"")
        end
        return nil
    end
    if (WasNotVisible) then
    shpfld.SetVisible(FALSE)
    end
    end
    if (MsgBox.YesNo("Add shapefile as theme to the view?", "Convert to Shapefile",true).Not) then
        continue
    end
    '****
    'Create a theme and add it to the View
    '****
    fthm = FTheme.Make(anFTab)
    theView.AddTheme(fthm)
    '****
    'Make the View Window Active
    '****
    theView.GetWin.Activate
end
'****
'Update the Legend in the View TOC
'****
fthm.InvalidateLegend
'****
'Draw the Theme
'****
fthm.SetVisible(true)
```

```
' Name:   Table.QueryFacilityShapefile
'
' Author:  Rod Schatz
'          University of Alberta
'          Edmonton, Alberta, 1998
'
' Date:    July 12th, 1998
'
' Title:  Query Facility Solution File
'
' Topics: MsgBox, List, Export, FTab, GetSelection, Update, Query
'
' Description:   This Avenue Script is one in a series of scripts used to create the A R C L A extension. The script will
query
'                a Theme based on a field labeled as “P”.  This field corresponds to the output file for the location-
allocation
'                program.  The *_facility_locations.txt file should have been joined to a an existing point Theme using a
'                one-to-many join.  This script is designed to be run on the Theme containing the join.  It will
breakdown
'                the solutions from the location-allocation program based on the P values.  This is required if the user
wishes
'                to create or draw the desire lines.
'
' Requires:
'
' Self:
'
' Returns:
'
' Updated:  January 3rd, 1999
'****
'Variable Initialization
'****
theView = av.GetActiveDoc

theTheme = theView.GetActiveThemes.Get(0)

'****
'Error Checking to ensure the chosen Theme has an attribute table (FTheme)
'****
if (theTheme.Is(FTheme).Not) then
  MsgBox.Error("Active theme is not FTheme", "Error")
  return nil
end
'****
'Create a List Box with "P" values
'****
theNumberList = { 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25 }
theUserChoice = MsgBox.ListAsString(theNumberList, "Please select a P value to extract from the database:", "Select
a P value ")
'****
'Error handling, script will end if the user does not select a P value
'****
if (theUserChoice = nil) then
    MsgBox.Warning("No selection made." + NL + "Program Cannot continue" + NL + "Press OK to EXIT",
"Warning")
    return nil
end
'****
'Get the Themes Feature Table
'****
theVTab = theTheme.GetFTab
```

```
'****
'Clear the highlighted selection, to ensure the user does not select
'multiple values of "P"
'****
theTheme.GetFTab.GetSelection.ClearAll
theTheme.GetFTab.UpdateSelection
theBitMap = theVTab.GetSelection
'****
'Find the Field labeled as "P"
'****
thePField = theVTab.FindField("P")
'****
'Error handling, if the field "P" is not present the script will end
'****
if (thePField = nil) then
    MsgBox.Error("This TABLE is not in the proper format for this script: script will end" + NL + "Program Cannot
continue" + NL + "Press OK to EXIT", "Error")
    exit
end
'****
'Set up and execute the Query
'****
theQueryString = "([P] = "+ theUserChoice.AsString +")"
theVTab.Query(theQueryString, theBitMap, #VTAB_SELTYPE_NEW)
theVTab.SetSelection(theBitMap)
'****
'Convert the selection to a new shapefile
'****
'MsgBox.YesNo("Would you like to convert the selection to a new shapefile?", "Convert to Shapefile", TRUE)
theConvertedFile = av.GetActiveDoc
for each t in theView.GetActiveThemes
    if (t.Is( FTHEME ).Not) then
        if (t.CanExportToFtab.Not) then
            continue
        end
    end
    if (t.Is (FTHEME).Not) then
        def = av.GetProject.MakeFileName("theme", "shp")
        def = FileDialog.Put(def, "*.shp", "Convert " + t.getName)
        if (def = NIL) then
            return NIL
        end
        anFTab = t.ExportToFtab(def)
    else
        tbl = t.GetFTab
        attribVis = FALSE
    for each f in tbl.GetFields
        if ((f.IsVisible) and not (f.IsTypeShape)) then
            attribVis = TRUE
            break
        end
    end
    shapeVis = tbl.FindField("Shape").IsVisible
    if ((attribVis and shapeVis).Not) then
        continue
    end
    def = av.GetProject.MakeFileName("theme", "shp")
    def = FileDialog.Put(def, "*.shp", "Convert " + t.getName)
    if (def = NIL) then return nil end
    shpfld = (tbl.FindField("Shape"))
    if (shpfld.IsVisible.Not) then
        shpfld.SetVisible(shpfld.IsVisible.Not)
```

```
            WasNotVisible = TRUE
        else
            WasNotVisible = FALSE
        end
        anFTab = tbl.Export(def, Shape, tbl.GetSelection.Count > 0)
        if (anFTab.HasError) then
            if (anFTab.HasLockError) then
                MsgBox.Error("Unable to acquire Write Lock for file " + def.GetBaseName,"")
            else
                MsgBox.Error("Unable to create " + def.GetBaseName,"")
            end
            return nil
        end
        if (WasNotVisible) then
            shpfld.SetVisible(FALSE)
        end
    end
    if (MsgBox.YesNo("Add shapefile as theme to the view?",
        "Convert to Shapefile",true).Not) then
        continue
    end
    '****
    'Create a theme and add it to the View
    '****
    fthm = FTheme.Make(anFTab)
    theView.AddTheme(fthm)
    '****
    'Make the View Window Active
    '****
    theView.GetWin.Activate
    end
    '****
    'Update the Legend in the View TOC
    '****
    fthm.InvalidateLegend
    '****
    'Draw the Theme
    '****
    fthm.SetVisible(true)
```

```
' Name:      Table.QueryNonZero
'
' Author:    Rod Schatz
'            Earth and Atmospheric Sciences Department
'            University of Alberta, 1999
'
' Date:  January 15th, 1999
'
' Title:  Query a Table for all values greater than 0
'
' Topics: List, Query, Selection, Theme, Ftab
'
' Description: This Avenue Script is one in a series of scripts used to create the A R C L A extension. Script will look
for the
'                active View and ask the user to select a Theme within the active View.  Next the script asks the use to
select
'                a Field that they would like to Query for all values greater than zero.  After the selection has been
created it
'                will prompt the user to convert the new selection to a shapefile.  Yes allows the process to continue, No
'                stops the script.  The reason we query for values greater than zero to remove all blanks from our tables.
'                This is performed because the location-allocation program cannot accept zero or blank values for the
'                demand      points file.  If zeros or blanks are included in the table it will significantly skew the final
solutions
'                from the location-allocation program.
'
' Requires:  An active View with one or more Themes present.
'
' Self:
'
' Returns:  A selection set for all values greater than zero.
'
' Updated:  February 20th, 1999
'****
' find the Active Document, must be a View
'****
theView = av.GetActiveDoc
'****
' Error Handling, to ensure active document is a View
'****
if (theView.Is(View).Not) then
    MsgBox.Warning("A View must be active (open)" + NL + "Program Cannot continue" + NL +  "Press OK to
EXIT", "Warning")
    return nil
end
'****
'Return all the Themes in the Active View
'****
theThemeList=theView.GetThemes
if (theThemeList.Count=0) then
    MsgBox.Error("There are no Themes in your View.", "Warning")
    return(nil)
end
'****
'Turn off all Themes
'****
for each theThemeList in theView.GetThemes
    if (theThemeList.IsActive) then
        theThemeList.SetActive(false)
    end
end
'****
' Create a message box to allow the user to select the themes to be used as the Facilities Point shape files
```

```
'****
theTheme = MsgBox.ListAsString(theThemeList, "Please select a Theme to be Queried","Select a Theme")
'****
' Warning Box if user does NOT select a Theme and script will Quit
'****
if (theTheme = nil) then
    MsgBox.Error("No theme selected." + NL + "Program Cannot continue" + NL + "Press OK to EXIT", "Error")
    exit
end
'****
'Set the "working" Theme active and de-select the other Themes
'****
theTheme.SetActive (TRUE)
'****
'Obtain the Themes FTab (Feature Table)
'****
theFTab = theTheme.GetFTab
'****
' Warning Box if user does NOT select a Theme and script will Quit
'****
if (nil = VTab) then
    MsgBox.Error("Unable to open Theme's Table"+ NL + "Program Cannot continue" + NL + "Press OK to EXIT",
"Error")
    exit
end
'****
'Find all the Fields in the chosen Theme's Table
'****
theFields = theFTab.GetFields
'****
'Create a dictionary to store the data from the previous command
'****
theDict = Dictionary.Make(theFields.Count)
for each i in theFields
    if (i.is(Field)) then
        theDict.Add(i.GetName,i)
    end
end
'****
'Display the Fields available in the table selected by the user
'****
theFieldList = MsgBox.List(theFTab.GetFields, "Select a Field to Query for all values > 0", "Field Selection")
'****
'Warning message if the user does not select any Fields the script will close
'****
if (theFieldList = nil) then
    MsgBox.Error("No Field selected." + NL + "Program Cannot continue" + NL + "Press OK to EXIT", "Error")
    return (nil)
end
'****
'Clear any selected records, this way the script will never overwrite any selection sets
'****
theTheme.GetFTab.GetSelection.ClearAll
theTheme.GetFTab.UpdateSelection
'****
'Now it is safe to obtain the FTab's Selection
'****
theBitmap = theFTab.GetSelection
'****
'Establish the Query Syntax (string) Statement
'****
theQuery = "([ "++ theFieldList.GetName ++" ] > 0.000000)"
```

```
'****
'Perform the Query and update selection
'****
theFTab.Query(theQuery, theBitmap, #VTAB_SELTYPE_NEW)
theFtab.UpdateSelection
'****
'Message box to ask the user if they would like to convert the selection
'to a new shapefile
'****
theUserChoice = MsgBox.YesNo("Convert the selection to a new shapefile?", "Convert Selection", TRUE)
'****
'Error handling if the user selects No, script Quits
'****
if (theUserChoice = FALSE) then exit end
'****
'ESRI's Convert to Shapefile script follows
'****
theView = av.GetActiveDoc
for each t in theView.GetActiveThemes
    p = FALSE
    if (t.Is( FTHEME ).Not) then
        if (t.CanExportToFtab.Not) then continue end
        def = av.GetProject.MakeFileName("theme", "shp")
        def = FileDialog.Put(def, "*.shp", "Convert " + t.getName)
        if (def = NIL) then return NIL end
            anFTab = t.ExportToFtab(def)
            ' For Database themes, which can return a nil FTab sometimes
            if (anFTab=nil) then
                MsgBox.Warning("Error occurred while converting to shapefile."+NL+ "Shapefile was not created.",
"Convert " + t.getName)
                continue
            end
            shpfld = anFTab.FindField("Shape")
    else
        tbl = t.GetFTab
        attribVis = FALSE
        for each f in tbl.GetFields
            if ((f.IsVisible) and not (f.IsTypeShape)) then
                attribVis = TRUE
                break
            end
        end
    shapeVis = tbl.FindField("Shape").IsVisible
    if ((attribVis and shapeVis).Not) then
        continue
    end
    def = av.GetProject.MakeFileName("theme", "shp")
    def = FileDialog.Put(def, "*.shp", "Convert " + t.getName)
    if (def = NIL) then return nil end
    shpfld = (tbl.FindField("Shape"))
    if (shpfld.IsVisible.Not) then
        shpfld.SetVisible(shpfld.IsVisible.Not)
        WasNotVisible = TRUE
    else
        WasNotVisible = FALSE
    end
    ' see if the view is projected
    thePrj = theView.GetProjection
    if (thePrj.IsNull.Not) then
        p = MsgBox.YesNoCancel("ArcView has detected that your view is projected."++ "Do you want the new
shapefile to be saved in the projected units?", "Convert", FALSE)
        if (p = Nil) then return nil end
        end
        ' if the user wants to project the shape, use ExportProjected
        if (p) then
            anFTab = tbl.ExportProjected(def, thePrj, tbl.GetSelection.Count > 0)
    else
        anFTab = tbl.Export(def, Shape, tbl.GetSelection.Count > 0)
    end
    if (anFTab.HasError) then
        if (anFTab.HasLockError) then
            MsgBox.Error("Unable to acquire Write Lock for file " +
    def.GetBaseName,"")
    else
            MsgBox.Error("Unable to create " + def.GetBaseName,"")
        end
        return nil
    end
    if (WasNotVisible) then
        shpfld.SetVisible(FALSE)
    end
    end
    ' build the spatial index
    anFTab.CreateIndex(shpfld)
    ' don't add the projected shapefile to the view - it won't show up if you do!
    if (p.Not) then
        if (MsgBox.YesNo("Add shapefile as theme to the view?", "Convert to Shapefile",true).Not) then
            continue
        end
    ' create a theme and add it to the View
        fthm = FTheme.Make(anFTab)
        theView.AddTheme(fthm)
    else
        MsgBox.Info("Since your view is projected, the projected data" ++ "will not be added to the view. It has been
saved on disk.", "Convert")
    end
    ' bring the View to the front
    theView.GetWin.Activate
end
```

```
' Name:    Theme.ChoroplethMap
'
' Author:  Rod Schatz
'          University of Alberta
'          Edmonton, Alberta, 1998
'
' Date:    January 1st, 1999
'
' Title:  Create Choropleth Map
'
' Topics:  MsgBox, List, Export, FTab, GetSelection, Update, Query
'
' Description: This Avenue Script is one in a series of scripts used to create the A R C L A extension. This script creates
a
'              choropleth map allowing the user to select the classification field variable, the normalize by field
variable
'              and the number of classes the user wishes to present the data in.  The script will automatically display
the
'              data using a quantile distribution.
'
' Requires:
'
' Self:
'
' Returns:
'
' Updated:  February 11th, 1999
'****
' Find the Active Document, must be a View
'****
theView = av.GetActiveDoc
'****
' Error Handling, to ensure active document is a View
'****
if (theView.Is(View).Not) then
    MsgBox.Warning("A View must be active (open)" + NL + "Program Cannot continue" + NL +
"Press OK to EXIT", "Warning")
    return nil
end
theTheme = theView.GetActiveThemes.Get(0)
'****
'Error Checking to ensure the chosen Theme has an attribute table (FTheme)
'****
if (theTheme.Is(FTheme).Not) then
    MsgBox.Error("Active theme is not FTheme", "Error")
    return nil
end
'****
'Error checking to ensure that the Theme is a Polygon Theme
'****
if ((theTheme.GetSrcName.GetSubName = "Polygon").Not) then
    MsgBox.Error("The chosen Theme must be Polygon Theme" + NL + "Program Cannot continue" + NL + "Press
OK to EXIT", "Error")
    return nil
end
'****
'Obtain the Themes FTab (Feature Table)
'****
theLegend = theTheme.GetLegend
theFTab = theTheme.GetFTab
flist = {}
for each f in theFTab.GetFields
```

```
    if ((f.IsTypeShape.Not) and (f.IsVisible)) then
        flist.Add(f)
    end
end
TextField = MsgBox.ListAsString(flist,"Select Field to display:","Select Field")
if (TextField = nil) then
    return nil
end
theNumbersList = {1,2,3,4,5,6,7,8,9,10}
theNumber = Msgbox.ListAsString(theNumbersList, "Select the number of Classes:", "Number of classes")
if (theNumber = nil) then exit end
'****
'Change the Legend to Graduated Symbol, Quantiles and 5 classes
'****
theLegend.SetLegendType(#LEGEND_TYPE_COLOR)
theLegend.Quantile(theTheme, TextField.GetName, theNumber)
theNormalizationField = MsgBox.ListAsString(flist,"Select Field to normalize the" + NL + "variable by:","Select
Normalization Field")
if (theNormalizationField = nil) then
    return nil
end
'****
'Normalize
'****
theNullValue = theLegend.GetNullValue (theNormalizationField.GetName)
theNullValue = theLegend.SetNullValue(theNormalizationField.GetName, -99)
theNullColor = Color.GetWhite
theNullcolor.SetTransparent(TRUE)
theLegend.SetNullValue(theNullColor.AsString, -99)
theLegend.SetNormType(#LEGEND_NORMTYPE_FIELD)
theLegend.SetNormFieldName(theNormalizationField.GetName)
'****
'Create a Colour Ramp Display
'****
theColourRamp = SymbolList.GetPredefined(#SYMLIST_TYPE_COLORRAMP).Get(32)
theLegend.GetSymbols.RampSavedColors(theColourRamp)
theTheme.InvalidateLegend
theTheme.UpdateLegend
'****
'Do it again to by-pass the bug
'****
theLegend.SetLegendType(#LEGEND_TYPE_COLOR)
theLegend.Quantile(theTheme, TextField.GetName, theNumber)
theLegend.SetNullValue(theNormalizationField.GetName, -99)
theLegend.SetNormType(#LEGEND_NORMTYPE_FIELD)
theLegend.SetNormFieldName(theNormalizationField.GetName)
theNullValue = theLegend.GetNullValue (theNormalizationField.GetName)
theNullValue = theLegend.SetNullValue(theNormalizationField.GetName, -99)
theNullValue = Color.GetWhite
theNullValue.SetTransparent(TRUE)
theLegend.SetNullValue(theNullValue.AsString, 0)
'****
'Create a Colour Ramp Display
'****
theColourRamp = SymbolList.GetPredefined(#SYMLIST_TYPE_COLORRAMP).Get(32)
theLegend.GetSymbols.RampSavedColors(theColourRamp)
theTheme.InvalidateLegend
theTheme.UpdateLegend
```

```
' Name: Theme.ConvertEAuIDtoNumber
'
' Author: Rod Schatz
'          Earth and Atmospheric Sciences Department
'          University of Alberta, 1999
'
' Date:  January 23rd, 1999
'
' Title:  Theme.Convert Field EAuID from Text to Number
'
' Topics: Calculate, FTab
'
' Description: This Avenue Script is one in a series of scripts used to create the A RC L A extension.  The EAUID field
is
'                 normally formatted as a text or string and the location-allocation program changes these values to
'                 numerical.  Consequently, the EAUID number needs to be converted to numerical format to facilitate a
'                 join between a solution file and a join.  Script adds a new Field called EA_Number to the active Theme.
'                 Then the script searches for a Field called EAuID and then converts it from text to a number format in
the
'                 new field.  The new field is required to join the output of the LA programs to a shapefile as the LA
program
'                 changes the formatting of the EAuID field.
'
' Requires:
'
' Self:
"
' Updated:
'****
'Find the active Document in the Project
'****
theView = av.GetActiveDoc
if (theView = Nil) then
   MsgBox.Warning("No active Views in your Project", "Error")
   exit
end
ThemeList = av.GetActiveDoc.GetActiveThemes
theTheme = ThemeList.Get(0)
theFTab = theTheme.GetFTab
theFTab.SetEditable(true)
theOldField = theFTab.FindField("EAuID")
if (theOldField = nil) then
   MsgBox.Error("Active Theme does not have the required EAuID field", "Error")
   exit
end
while (true)
if (theFTab.FindField("EA_Number") = nil) then
   theNewField = Field.Make("EA_Number", #FIELD_LONG, 16, 0)
   theFTab.AddFields({theNewField})
   theExpression = "([EAUID].AsNumber)"
   theFTab.Calculate(theExpression, theNewField)
   theFTab.SetEditable(false)
else
   break
   end
end
theFtab.Refresh
```

```
' Name:   Theme.GraduatedSymbolMap
'
' Author:  Rod Schatz
'          University of Alberta
'          Edmonton, Alberta, 1998
'
' Date:    January 10th, 1999
'
' Title:  Create Graduated Symbol Point Map
'
' Topics:  View, Theme, Graduated Symbol Map
'
' Description: This Avenue Script is one in a series of scripts used to create the A R C L A extension. The script will
create a
'              Graduated Symbol Map for an active Point Theme in a View.  The script will create a graduated symbol
map
'              for a variable chosen by the user and display the chosen variable using a quantile distribution with the
number
'              of classes selected by the user.
'
' Requires:
'
' Self:
'
' Returns:
'
' Updated:  January 30th, 1999
'****
'Find Project's active View
'****
theView = av.GetActiveDoc
theTheme = theView.GetActiveThemes.Get(0)
'****
'Error Checking to ensure the chosen Theme has an attribute table (FTheme)
'****
if (theTheme.Is(FTheme).Not) then
    MsgBox.Error("Active theme is not FTheme", "Error")
    return nil
end
'****
'Error Checking to ensure the chosen Theme is a point Theme
'****
if ((theTheme.GetSrcName.GetSubName = "point").Not) then
    MsgBox.Error("The chosen Theme must be a point theme" + NL + "Program Cannot continue" + NL +
"Press OK to EXIT", "Error")
    return nil
end
'****
'Get the Theme's Legend
'****
theLegend = theTheme.GetLegend
theFTab = theTheme.GetFTab
flist = {}
for each f in theFTab.GetFields
    if ((f.IsTypeShape.Not) and (f.IsVisible)) then
        flist.Add(f)
    end
end
TextField = MsgBox.ListAsString(flist,"Select Field to display:","Select Field")
if (TextField = nil) then
    return nil
end
```

```
theNumbersList = {1,2,3,4,5,6,7,8,9,10}
theNumber = Msgbox.ListAsString(theNumbersList, "Select the number of Classes:", "Number of classes")
if (theNumber = nil) then exit end
'****
'Change Legend to Graduated Symbol with a Quantile distribution with 5 classes
'****
theLegend.SetLegendType(#LEGEND_TYPE_SYMBOL)
theLegend.Quantile(theTheme, textfield.getname, theNumber)
theLegend.GetSymbols.RampSizes(8,16)
theColorRamp = SymbolList.GetPreDefined(#SYMLIST_TYPE_COLORRAMP).Get(32)
theLegend.GetSymbols.RampSavedColors(theColorRamp)
'****
'Refresh the View and make the Changes
'****
theTheme.InvalidateLegend
theTheme.UpdateLegend
```

```
' Name: Theme.LegendFacilityUniqueSymbol
'
' Author: Rod Schatz
'          Earth and Atmospheric Sciences Department
'          University of Alberta, 1997
'
' Date:   December 5th, 1997
'
' Title: Change Legend To Graduated Symbol
'
' Topics:
'
' Description: This Avenue Script is one in a series of scripts used to create the A R C L A extension.
'                    Changes the legend symbols for a facility solution file to unique symbols.
'
'
' Requires:
'
' Self:
'
' Returns: A new legend display
'
' Updated: January 29th, 1999
theView = av.GetActiveDoc
theTheme = theView.GetActiveThemes.Get(0)
'****
'Error Checking to ensure the chosen Theme has an attribute table (FTheme)
'****
if (theTheme.Is(FTheme).Not) then
    MsgBox.Error("Active theme is not FTheme", "Error")
    return nil
end
'****
'Error Checking to ensure the chosen Theme is a point Theme
'****
if ((theTheme.GetSrcName.GetSubName = "point").Not) then
    MsgBox.Error("The chosen Theme must be a point theme" + NL + "Program Cannot continue" + NL + "Press OK
to EXIT", "Error")
    return nil
end
theField = "Facility"
'****
'Error Handling to ensure the Facility Field exists
'****
if (theField = nil) then exit end
theLegend = theTheme.GetLegend
theFTab = theTheme.GetFTab
flist = {}
for each f in theFTab.GetFields
    if ((f.IsTypeShape.Not) and (f.IsVisible)) then
        flist.Add(f)
    end
end
TextField = MsgBox.ListAsString(flist,"Select Field to display:","Select Field")
if (TextField = nil) then
    return nil
end
'****
'Change Legend to Graduated Symbol
'****
theLegend.SetLegendType(#LEGEND_TYPE_UNIQUE)
theLegend.Unique(theTheme, TextField.GetName)
theColorRamp = SymbolList.GetPreDefined(#SYMLIST_TYPE_COLORRAMP).Get(32)
```

```
theLegend.GetSymbols.RandomSymbols
theTheme.UpdateLegend
```

```
' Name: Theme.LegendMakeSimple
'
' Author: Rod Schatz
'            Earth and Atmospheric Sciences Department
'            University of Alberta, 1999
'
' Date:   December 7th, 1997
'
' Title: Make a simple legend
'
' Topics:  Theme, Legend
'
' Description: This Avenue Script is one in a series of scripts used to create the A R C L A extension.  Sets
'                   the Legend to a Single Grey Symbol Legend
'
' Requires:
'
' Self:
'
' Returns:
'
' Updated: February 12th, 1999
'****
' Get the view, theme and finally the legend for the theme.
'****
theView = av.GetActiveDoc
theTheme = theView.GetActiveThemes.Get(0)
theLegend = theTheme.GetLegend
theLegend.SingleSymbol
'****
' Change the legend's color by first getting the symbols, which returns a symbol list, and then changing color
'of the first (only) element in that list to gray.
'****
theLegend.GetSymbols.Get(0).SetColor(Color.GetGray)
'****
' Update the theme with the changes made to its legend.  This is analogous to clicking the Apply button.
'****
theTheme.InvalidateLegend
theTheme.UpdateLegend
```

```
' Name: View.AddXYCoordToFTab
'
' Author: ESRI
'
' Modified by: Rod Schatz
'                 Earth and Atmospheric Sciences Department
'                 University of Alberta, 1998
'
' Title:  Adds X and Y coordinates of features to Attribute Table
'
' Topics:  GeoData
'
' Description:   This Avenue Script is one in a series of scripts used to create the A R C L A extension. Adds two new
fields,
'                 named X-coord and Y-coord, to the table of the first active theme in the TOC and fills the respective
fields
'                 with the X,Y coordinates of the selected points (or all points if no selection is defined) in a point theme.
If
'                 instead the active theme is a polygon theme, then the X,Y coordinates of the polygon centroid are
'                 calculated. If the theme is projected, the output coordinates will also be projected.
'
' Requires:  An active point or polygon theme. This script does minimal error checking and assumes that an active
theme.
'
' Self:
'
' Returns:
'****
'Inform the user that this script will only calculate the correct values for latitude and longitude if it is not projected
'****
theUserAnswer = MsgBox.YesNo("Script will calculate the Lat. and Long. for a Theme." + NL + "It must be
calculated on unprojected data." + NL + "Select YES if your data is unprojected and you would like to continue?" + NL
+ "Select NO to change the projection in the View - Properties box", "Continue",TRUE)
'****
'Error Handling if the user selects NO
'****
if (theUserAnswer.Not) then exit end
'****
'Get the Active Document
'****
theView = av.GetActiveDoc
'****
'Error Handling to ensure the user has a View open as the active document
'****
if (theView.Is(View).Not) then
    MsgBox.Warning("A View must be active (open)" + NL + "Program Cannot continue" + NL +  "Press OK to
EXIT", "Warning")
    return nil
end
'****
'must be global to work in Calc exp below
'****
_theProjection = theView.GetProjection
project_flag = _theProjection.IsNull.Not  'true if projected
theTheme = theView.GetActiveThemes.Get(0)
'****
'Check if point or polygon theme
'****
if (((theTheme.GetSrcName.GetSubName = "point") or (theTheme.GetSrcName.GetSubName = "polygon")).Not) then
    MsgBox.Info("Active theme must be polygon or point theme","")
    exit
end
```

```
'****
'get the theme table and current edit state
'****
theFTab = theTheme.GetFTab
theFields = theFTab.GetFields
edit_state = theFTab.IsEditable
'****
'make sure table is editable and that fields can be added
'****
if (theFtab.CanEdit) then
    theFTab.SetEditable(true)
    if ((theFTab.CanAddFields).Not) then
        MsgBox.Info("Can't add fields to the table."+NL+"Check write permission.", "Can't add X,Y coordinates")
        exit
    end
else
    MsgBox.Info("Can't modify the feature table."+NL+ "Check write permission.","Can't add X,Y coordinates")
    exit
end
'****
'Check if fields named "X-coord" and Y-coord" exist
'****
x_exists = (theFTab.FindField("X-coord") = NIL).Not
y_exists = (theFtab.FindField("Y-coord") = NIL).Not
if (x_exists or y_exists) then
    if (MsgBox.YesNo("Overwrite existing fields?", "X-coord, Y-coord fields already exist", false)) then
        '****
        'if ok to overwrite, delete the fields as they may not be defined
        'as required by this script (eg., created from another script).
        '****
        if (x_exists) then
            theFTab.RemoveFields({theFTab.FindField("X-coord")})
        end
        if (y_exists) then
            theFTab.RemoveFields({theFTab.FindField("Y-coord")})
        end
    else
        exit
    end
end
x = Field.Make ("X-coord",#FIELD_DECIMAL,18,5)
y = Field.Make ("Y-coord",#FIELD_DECIMAL,18,5)
theFTab.AddFields({x,y})
'****
'Get point coordinates or polygon centroid coordinates
'****
if (theTheme.GetSrcName.GetSubName = "point") then
    if (project_flag) then
        'Projection defined
        theFTab.Calculate("[Shape].ReturnProjected(_theProjection).GetX", x)
        theFTab.Calculate("[Shape].ReturnProjected(_theProjection).GetY", y)
    else
        'No projection defined
        theFTab.Calculate("[Shape].GetX", x)
        theFTab.Calculate("[Shape].GetY", y)
    end
else
'****
'polygon case
'****
    if (project_flag) then
        theFTab.Calculate("[Shape].ReturnCenter.ReturnProjected(_theProjection).GetX", x)
        theFTab.Calculate("[Shape].ReturnCenter.ReturnProjected(_theProjection).GetY", y)
    else
        theFTab.Calculate("[Shape].ReturnCenter.GetX", x)
        theFTab.Calculate("[Shape].ReturnCenter.GetY", y)
    end
end
'Return editing state to pre-script running state
theFTab.SetEditable(edit_state)
```

```
' Name:  View.CalculateFeatureLength
'
' Author: ESRI
'
' Modified By: Rod Schatz
'                    Department of Earth and Atmospheric Sciences
'                    University of Alberta, 1999
'
' Title:  Calculates feature geometry values
'
' Topics:
'
' Description:   his Avenue Script is one in a series of scripts used to create the A R C L A extension. Calculates the
length
'                    line theme. . The calculation is stored in a new field.  The script will add the field:  Length to line
themes  if
'                    do not exist. If the field exist the values wll be recalculated. Rerun the script if you change the
projection of
'                    iew.
'
' Requires:  A View with at least one active theme.  You must have write access to the active theme(s).
'
' Self:
'
' Returns:
'
' Updated
'****
'Inform the user that their View must be Projected
'****
theUserAnswer =
MsgBox.YesNo("The View must be Projected in order to properly" + NL + "calculate length on your Line Theme." +
NL + "Select YES if your data is Projected." + NL + "Select NO to change the projection in the View - Properties box",
"Continue",TRUE)
'****
'Error Handling if the user selects NO
'****
if (theUserAnswer.Not) then exit end
'****
' Get the view and its projection if any.
'****
theView = av.GetActiveDoc
'****
'set Map Units to feet and Distance Units to miles
'****
theView.SetUnits(#UNITS_LINEAR_METERS)
theView.GetDisplay.SetDistanceUnits(#UNITS_LINEAR_METERS)
thePrj = theView.GetProjection
if (thePrj.IsNull) then
    hasPrj = false
else
    hasPrj = true
end
'****
' Get the list of active themes. if there aren't any, let the user know and exit.
'****
theActiveThemeList = theView.GetActivethemes
if (theActivethemeList.Count = 0) then
    MsgBox.Error("No active themes.","")
    Exit
end
'****
```

```
' Loop through the list of active themes. if you can't edit the theme, inform the user.
'****
For Each thetheme in theActivethemeList
    theFTab = thetheme.GetFTab
    if (theFTab.CanEdit.Not) then
        MsgBox.Info("Cannot edit table for theme:"++thetheme.AsString,"")
        Continue
    end
    '****
    ' Make the FTAB editable, and find out which type of feature it is.
    '****
    theFTab.SetEditable(TRUE)
    theType = theFTab.FindField("shape").GetType
    '****
    'Error Handling to ensure Theme is a Polygon
    '****
    if ((theType = #FIELD_SHAPELINE).Not) then
        MsgBox.Error("Your Theme must be a Line Theme"+ NL + "Program Cannot continue" + NL + "Press OK to
EXIT", "Error")
        exit
    end
    if (theType = #FIELD_SHAPELINE) then
    '****
    ' if the data source is linear, check for the existence of the  field "Length". if it doesn't exist, create it.
    '****
        if (theFTab.FindField("Length") = nil) then
            theLengthField = Field.Make("Length",#FIELD_DOUBLE,16,3)
            theFTab.AddFields({theLengthField})
        else
            ok = MsgBox.YesNo("Update Length?", "Calculate", true)
                if (ok.Not) then
                    continue
                end
            theLengthField = theFTab.FindField("Length")
        end
    '****
    'Loop through the FTAB and find the projected length of each shape and set the field values appropriately.
    '****
        theShape = theFTab.ReturnValue(theFTab.FindField("shape"),0)
        For Each rec in theFTab
            theFTab.QueryShape(rec,thePrj,theShape)
            theLength = theShape.ReturnLength
            theFTab.SetValue(theLengthField,rec,theLength)
        end
    end
end
theFTab.SetEditable(FALSE)
```

```
' Name:  View.CalculatePolygonGeometry
'
' Author:  ESRI
'
' Modified By: Rod Schatz
'                    Department of Earth and Atmospheric of Sciences
'                    University of Alberta, 1999
'
' Date:  January 28th, 1999
'
' Title:  Calculates Polygon Geometry Values
'
' Topics:
'
' Description:   This Avenue Script is one in a series of scripts used to create the A R C L A extension. Calculates area
and
'                     perimeter for polygon themes.  If the View has been projected the calculations are in projected meters.
The
'                     script will add the fields: Area and Perimeter to polygon themes. If the fields exist, their values will be
'                     recalculated. Rerun the script if you change the projection of the view.
'
' Requires:  A View with at least one active theme.  You must have write access to the active theme(s).
'
' Self:
'
' Returns:
'
' Updated:
'****
'Inform the user that their View must be Projected
'****
theUserAnswer =
MsgBox.YesNo("The View must be Projected in order to properly" + NL + "calculate Area, Perimeter on your Polygon
Theme." + NL + "Select YES if your data is Projected." + NL + "Select NO to change the projection in the View -
Properties box", "Continue",TRUE)
'****
'Error Handling if the user selects NO
'****
if (theUserAnswer.Not) then exit end
'****
' Get the view and it's projection if any.
'****
theView = av.GetActiveDoc
thePrj = theView.GetProjection
if (thePrj.IsNull) then
    hasPrj = false
else
    hasPrj = true
end
'****
' Get the list of active themes. If there aren't any, let the user know and exit.
'****
theActivethemeList = theView.GetActivethemes
if (theActivethemeList.Count = 0) then
    MsgBox.Error("No Active Themes.", + NL + "Program Cannot continue" + NL + "Press OK to EXIT", "Error")
    Exit
end
'****
' Loop through the list of active themes. If you can't edit the theme, inform the user.
'****
For Each thetheme in theActivethemeList
    theFTab = thetheme.GetFTab
    if (theFTab.CanEdit.Not) then
        MsgBox.Info("Cannot edit table for Theme:"++thetheme.AsString, "Warning")
        Continue
    end
    '****
    ' Make the FTAB editable, and find out which type of feature it is.
    '****
    theFTab.SetEditable(TRUE)
    theType = theFTab.FindField("shape").GetType
    '****
    'Error Handling to ensure Theme is a Polygon
    '****
    if ((theType = #FIELD_SHAPEPOLY).Not) then
        MsgBox.Error("Your Theme must be a Polygon Theme"+ NL + "Program Cannot continue" + NL + "Press OK
to EXIT", "Error")
        exit
    end
    if (theType = #FIELD_SHAPEPOLY) then
        '****
        ' If it's polygonal check for the existence of the fields "Area" and
        ' Perimeter. If they do not exist, create them.
        '****
        if (theFTab.FindField("Area") = nil) then
            theAreaField = Field.Make("Area",#FIELD_DOUBLE,16,3)
            theFTab.AddFields({theAreaField})
        else
            ok = MsgBox.YesNo("Update Area?", "Calculate", true)
                if (ok.Not) then
                    continue
                end
            theAreaField = theFTab.FindField("Area")
        end
        if (theFTab.FindField("Perimeter") = nil) then
            thePerimeterField = Field.Make("Perimeter",#FIELD_DOUBLE,16,3)
            theFTab.AddFields({thePerimeterField})
        else
            ok = MsgBox.YesNo("Update Perimeter?", "Calculate", true)
            if (ok.Not) then
                continue
            end
            thePerimeterField = theFTab.FindField("Perimeter")
        end
        '****
        'Loop through the FTAB and find the projected area and perimeter of each shape and set the field
        'values appropriately.
        '****
        theShape = theFTab.ReturnValue(theFTab.FindField("shape"),0)
        For Each rec in theFTab
            theFTab.QueryShape(rec,thePrj,theShape)
            theArea = theShape.ReturnArea
            thePerimeter = theShape.ReturnLength
            theFTab.SetValue(theAreaField,rec,theArea)
            theFTab.SetValue(thePerimeterField,rec,thePerimeter)
        end
    end
    theFTab.SetEditable(FALSE)
end
```

```
' Name: View.DisplaySelectedThemes
'
' Author: Rod Schatz
'         Earth and Atmospheric Sciences Department
'         University of Alberta, 1998
'
' Date:   December 11th, 1998
'
' Title: Displays/sets active the themes chosen by the user
'
' Topics:  View, Theme, Table of Contents
'
' Description: This Avenue Script is one in a series of scripts used to create the A R C L A extension.  Allows the user
to
'              select the Themes they want to turn on in that View.  Thus, allowing the user to turn on more than on
Theme
'              at once.  This replaces the escape key on the keyboard which has been disabled with ArcView 3.1.
'
' Requires:  A View with Themes in an ArcView Project
'
' Self:
'
' Returns:
'
' Updated:
'***
'Variable initialization
'****
theTitle = "Display on Selected Themes Script"
'****
' Get the ArcView Project Document
'****
theProject = av.GetProject
'****
' Select the View Document
'****
theDocList = theProject.GetDocs
'****
'Create a dictionary to store the data from the previous command
'****
theDict = Dictionary.Make(theDocList.Count)
for each i in theDocList
    if (i.is(View)) then
        theDict.Add(i.GetName,i)
    end
end
theList = theDict.ReturnKeys
theView = theDict.Get(MsgBox.ListAsString(theList, "Select your View", theTitle))
'****
' Error Handling - Warning Box if user does NOT select a View the script will Quit
'****
if (theView = nil) then
    MsgBox.Error("No View selected, Program Closing." , "Warning")
    Exit
end
'****
'Get the View's Themes
'****
if (theView.GetThemes.Count=0) then
    MsgBox.Error("No Themes in your View.","Warning")
    return(nil)
end
```

```
theThemeList=theView.GetThemes
if (theThemeList.Count=0) then
    MsgBox.Error("There are no Themes in your View.", "Warning")
    return(nil)
end
'****
'Turn off all Themes
'****
for each theThemeList in theView.GetThemes
    if (theThemeList.IsVisible) then
        theThemeList.SetVisible(false)
    end
end
'****
'Prompt user for a selection - Themes to set visible
'****
theThemes = MsgBox.MultiListAsString(theThemeList, "Select the Theme(s) you would like to set visible:" + NL +
"Use the Shift Key to select multiple Themes", "Select Theme(s)")
'****
'Error Handling - Warning Box if user does NOT select a Theme the script will Quit
'****
if (theThemes = nil) then
    MsgBox.Error("No Theme(s) selected, Program Closing." , "Warning")
    Exit
end
'****
'Turn on the users selection set
'****
for each theThemes in theThemes
    if (theThemes.IsVisible.not) then
        theThemes.SetVisible(TRUE)
    end
end
'****
'Force ArcView to Refresh/Redraw the View
'****
theView.InvalidateTOC(nil)
theView.GetDisplay.Invalidate(true)
```

```
' Name:  View.HideLegendForInvisibleThemes
'
' Author: ESRI
'
' Modified By: Rod Schatz
'                        Department of Earth and Atmospheric Sciences
'                        University of Alberta, 1999
'
' Date:   January 29th, 1999
'
' Title:  Hides the legend for all non visible themes
'
' Topics:  Legends
'
' Description: This Avenue Script is one in a series of scripts used to create the A R C L A extension. This script
'                        automatically hides/shows theme legends depending on whether or not the theme is visible. When a
theme
'                        is drawn, the theme legend will become visible, when a theme is toggled off the legend will be hidden.
'
' Requires:  The active document must be a view. This script must be installed as the update event on a
'                    control, or as the update script for the view document.
'
' Self: If this is an update script on the document, SELF is the View. If it is an update script on a control, SELF is the
'          control.
'
' Returns:
'
' Updated:
theThemes = av.GetActiveDoc.GetThemes
'****
'Get the current state
'****
state = av.GetActiveDoc.GetVisibleThemes.Count
'****
'Determine if the state has changed
'****
savedState = self.GetObjectTag
if ((savedState <> nil) and (savedState = state)) then
    exit
end
'****
'Save the current state
'****
SELF.SetObjectTag( state )
'****
'Update the TOC
'****
for each t in theThemes
    t.SetLegendVisible( t.IsVisible )
end
av.GetActiveDoc.InvalidateTOC( NIL )
```

```
' Name:    View.LocationAllocationDiagram
'
' Author: Rod Schatz and in part Iain Getty
'            Earth and Atmospheric Sciences Department
'            University of Alberta, 1997
'
' Date:    November 12, 1997
'
' Title:   Location-Allocation Spider Line Diagram
'
' Topics:  Table, VTab, Themes, PolyLines, View, Projection, TOC, Fields
'
' Description: This Avenue Script is one in a series of scripts used to create the A R C L A extension. This script is
designed
'                 to use two Themes one, which is designated as the demand points (created from the final_demand.txt file
from
'                 the LA program) theme and the other which is the create/allocate them to location of the corresponding
'                 facilities (created from the facility locations.txt file). The script will connect the demands and facilities
through
'                 desire lines. In order for the script to work, the script requires that both Themes have field labeled  facility
'                 which indicates which demand point is allocated to which facility location. The script also requires a field
'                 in the demand table called weight.
'
' Requires: A View with two point shape files. One to be used as the facility and the other as demand
'                 points. Both point shape files need a field called "Facility". The View should also have no
'                 Projection associated to it as the script will automatically specify the proper Projection variables.
'
' Self:
'
' Returns: A new theme drawing lines from the facility points to the demand points. Then reorders the TOC
'                to ensure the following order points, lines, polygons.
'
' Updated: March 15th, 1999
'****
' Variable Initialization
'****
theView = av.GetActiveDoc
'****
'Error Handling to ensure the user has a View open as the active document
'****
if (theView.Is(View).Not) then
    MsgBox.Warning("A View must be active (open)" + NL + "Program Cannot continue" + NL + "Press OK to
EXIT", "Warning")
    return nil
end
'****
' Creates another variable to get the themes from the TOC (Table of Contents)
'****
theThemesList = theView.GetThemes
'****
' Create a message box to allow the user to select the themes to be used as the Facilities Point shape files
'****
theFacilityTheme = MsgBox.ListAsString(theThemesList, "Please select the Theme to be used as the Facility Points",
"Select a Theme")
'****
' Warning Box if user does NOT select a theme and script will Quit
'****
if (theFacilityTheme = nil) then
    MsgBox.Error("No theme selected." + NL + "Program Cannot continue" + NL + "Press OK to EXIT", "Error")
    exit
end
'****
```

```
' Create a message box to allow the user to select the themes to be used as the Demand Point shape files
'****
theDemandPointTheme = MsgBox.ListAsString(theThemesList, "Please select the Theme to be used as the Demand
Points", "Select a Theme")
'****
' Warning Box if user does NOT select a theme and script will Quit
'****
if (theDemandPointTheme = nil) then
    MsgBox.Error("No theme selected." + NL + "Program Cannot continue" + NL + "Press OK to EXIT", "Error")
    exit
end
theViewProjection = theView.GetProjection
'****
'Variables to be used as our themes - theFacilityTheme and Demands
'****
theTheFacilityTheme = theView.FindTheme(TheFacilityTheme.AsString)
theDemand = theView.FindTheme(theDemandPointTheme.AsString)
'****
' Create a Feature Table
'****
theDemandFTab = theDemand.GetFTab
theTheFacilityThemeFTab = theTheFacilityTheme.GetFTab
'****
' Find the shape field of our two themes - should be points
'****
TheDemandCenter = theDemandFTab.FindField("Shape")
TheFacilityCenter = theTheFacilityThemeFTab.FindField("Shape")
theFacilityFieldList = theTheFacilityThemeFTab.GetFields
'****
' Message Boxes for user to select the Fields that correspond to their Facility EA numbers and Demand EA
'numbers
'****
theDemandPointsFieldList = theDemandFTab.GetFields
TheCenterFacilityField = theTheFacilityThemeFTab.FindField("Facility".AsString)
ThePointFacilityField = theDemandFTab.FindField("Facility".AsString)
'****
'Makes a list of the values for each record in the Demand Table and compares it 'the Facilities Table to see
'which demand point corresponds to which facility point.
'****
thePointList = List.Make
for each rec in theDemandFTab
    theDemandPointThemeoint = theDemandFTab.ReturnValue(TheDemandCenter,rec)
    theFacilityLink = theDemandFTab.ReturnValue(ThePointFacilityField,rec)
    for each s in theTheFacilityThemeFTab
        theFacilityPoint = theTheFacilityThemeFTab.ReturnValue(TheFacilityCenter,s)
        theFacilityID = theTheFacilityThemeFTab.ReturnValue(TheCenterFacilityField,s)
        if (theFacilityLink = theFacilityID) then
            thePointList.Add(theDemandPointThemeoint)
            thePointList.Add(theFacilityPoint)
        end
    end
end
'****
'Creates a new theme with one row (polyline) and draws the respective lines from the facilities to the
'demand points
'****
theTableClass = PolyLine
def = av.GetProject.MakeFileName("theme","shp")
def = FileDialog.Put(def,"*.shp","New Theme")
if (def <> nil) then
    theNewTable = FTab.MakeNew(def, theTableClass)
    theFirstNewField = Field.Make("Demand_No", #FIELD_LONG, 8, 0)
    theFirstNewField2 = Field.Make("Weight", #FIELD_LONG, 8, 0)
    theFirstNewField.SetVisible(TRUE)
    theNewTable.AddFields({theFirstNewField, theFirstNewField2})
    SpiderTheme = FTheme.Make(theNewTable)
    theView.AddTheme(SpiderTheme)
    SpiderTheme.SetActive(TRUE)
    SpiderTheme.SetVisible(FALSE)
    theView.SetEditableTheme(SpiderTheme)
    av.GetProject.SetModified(TRUE)
end
theCountList = thePointList.Count
Counter = 0
for each num in 1..theCountList by 2
    Counter = Counter + 2
    TheDemandPointThemeoint = thePointList.Get(num -1)
    TheFacilityPoint = thePointList.Get(num)
    theTheme = theView.GetEditableTheme
    spiderline = PolyLine.Make({{theDemandPointThemeoint,theFacilityPoint}})
    theTheme = theView.GetEditableTheme
    if (spiderline <> nil) then
        if (SpiderTheme <> nil) then
            theField = SpiderTheme.GetFTab.FindField("Shape")
            rec = SpiderTheme.GetFTab.AddRecord
            SpiderTheme.GetFTab.SetValue(theField, rec, spiderline)
            SpiderTheme.GetFTab.GetSelection.ClearAll
            SpiderTheme.GetFTab.GetSelection.Set(rec)
            SpiderTheme.GetFTab.UpdateSelection
    else
            gl = GraphicShape.Make(spiderline)
            theView.GetGraphics.UnselectAll
            gl.SetSelected(TRUE)
            theView.GetGraphics.Add(gl)
        end
        av.GetProject.SetModified(TRUE)
    end
end
SpiderTheme.SetVisible(TRUE)
'****
'Start copy weight field from demand file
'****
theNewVTab = SpiderTheme.GetFtab
theNewVTab.SetEditable (True)
theTheme.SetActive(false)
' ****
' Get the values from the demand field
' ****
theDemandFTab = theDemand.GetFTab
theOldField = theDEmandFTab.FindField("weight")
if (nil = theOldField) then
    MsgBox.Error ("Unable to find the EAuID field","Error")
    exit
end
'****
' The record count is required for a loop.
'****
theRecs = theDemandFTab.GetSelection
theBitmap = Bitmap.Make(theRecs.GetSize)
Numrecs=theDemandFtab.GetNumRecords
'****
' Read and write each record.
'****
for each index in theDemandFTab
```

```
    fieldValue = theDemandFtab.ReturnValueNumber (theOldField,index)
    theNewVTab.SetValue (theFirstNewField2, index, fieldValue)
end
theNewVTab.Flush
'****
' Stop editing the table
'****
SpiderTheme.StopEditing(TRUE)
SpiderTheme.SetActive(True)
'****
'Clear the last selected row in the new shape file and all other selected Themes.
'****
for each t in theView.GetActiveThemes
    t.ClearSelection
end
av.PurgeObjects
'****
' Now we want to change the order of the themes in the TOC (Table of Contents) to have the new theme at
'the bottom for better display purposes.
'****
theView = av.GetActiveDoc
theThemes = theView.GetThemes
if (theThemes = NIL) then
    exit
end
'****
' Theme types. Change the order of this list to change the display order in the TOC; images, and anything
'else not included in the list, are included in "other" category.
'****
theTypes = {"point","multipoint","labelpoint","node","annotation","arc", "multiline", "route", "polygon", "poly",
"region"}
'****
'Use a Dictionary to store themes of various types
'****
aDictionary = Dictionary.Make(theTypes.Count + 1)
for each type in theTypes
    aDictionary.Add (type,{})
end
aDictionary.Add("other",{})  ' Store types not in theTypes, eg images
'****
' Sort themes by type
'****
for each thm in theThemes
    aKey = thm.GetSrcName.GetSubName
    if (theTypes.FindByValue(aKey) >= 0) then
    '****
    ' Get the list from of dictionary and add theme to list
    '****
        aValue = aDictionary.Get(aKey) + {thm}
        aDictionary.Set(aKey,aValue)
    else
        '****
        ' Regions and routes are handled separately because their SubName also contains another component
        'besides the word "region" or "route".
        '****
        type = "other"  'Set default
        if (aKey.Contains("region")) then
            type = "region"
        end
        if (aKey.Contains("route")) then
            type = "route"
        end
        aValue = aDictionary.Get(type) + {thm}
        aDictionary.Set(type, aValue)
    end
end
'****
' Re-order the TOC by rebuilding the TOC list
'****
theThemes.Empty
for each type in theTypes
    theThemes.Merge(aDictionary.Get(type))
end
theThemes.Merge(aDictionary.Get("other"))
'****
'Refresh TOC and View
'Invalidate and refresh the View's Display
'****
theView.InvalidateTOC(nil)
theView.Invalidate
'****
'end re-order script
'Start the set Projection/Units Script
'****
theView = av.GetActiveDoc
'****
'set Map Units to feet and Distance Units to miles
'****
theView.SetUnits(#UNITS_LINEAR_METERS)
theView.GetDisplay.SetDistanceUnits(#UNITS_LINEAR_METERS)
'****
'Start the calculate length of the desire lines script
'****
thePrj = theView.GetProjection
if (thePrj.IsNull) then
    hasPrj = false
else
    hasPrj = true
end
'****
' Get the list of active themes. if there are none, let the user know and exit.
'****
theActivethemeList = theView.GetActivethemes
if (theActivethemeList.Count = 0) then
    MsgBox.Error("No active themes.","")
    Exit
end
'****
' Loop through the list of active themes. if you can't edit the theme inform the user.
'****
For Each thetheme in theActivethemeList
    theFTab = theTheme.GetFTab
    if (theFTab.CanEdit.Not) then
        MsgBox.Info("Cannot edit table for theme:"++thetheme.AsString,"")
        Continue
    end
    '****
    'Make the FTAB editable, and find out which type of feature it is.
    '****
    theFTab.SetEditable(TRUE)
    theType = theFTab.FindField("shape").GetType
    '****
    ' if the data source is linear, check for the existence of the  field "Length". if it doesn't exist, create it.
    '****
    if (theFTab.FindField("Length") = nil) then
        theLengthField = Field.Make("Length",#FIELD_DOUBLE,16,3)
        theFTab.AddFields({theLengthField})
    else
        theLengthField = theFTab.FindField("Length")
    end
    '****
    ' Loop through the FTAB and find the projected length of each shape and set the field values
    'appropriately.
    ' ****
    theShape = theFTab.ReturnValue(theFTab.FindField("shape"),0)
    For Each rec in theFTab
        theFTab.QueryShape(rec,thePrj,theShape)
        theLength = theShape.ReturnLength
        theFTab.SetValue(theLengthField,rec,theLength)
    end
end
'****
'Start Add Number Script
'Now Add uniwue number to each row
'****
theThemeList=theView.GetThemes
'****
'Turn off all Themes
'****
for each theThemeList in theView.GetThemes
    if (theThemeList.IsActive) then
        theThemeList.SetActive(false)
    end
end
'****
'Turn on the Desire Lines Theme
'****
SpiderTheme.SetActive(true)
numRecs = theFTab.GetNumRecords
'****
'Find the Field to the Table
'****
theNewField = theFTab.FindField("Demand_No")
'****
'Return each Row in the Table and assign it a number (incremented)
'****
for each recNum in theFTab
    theNumber = (recNum + 1).SetFormat("d").AsString
    theFTab.SetValue(theFirstNewField, recNum, theNumber)
end
'****
'Tidy Things up
'****
theFTab.SetEditable(FALSE)
theView.InvalidateTOC(nil)
theView.Invalidate
```

```
' Name: View.CreateNewView
'
' Author: Rod Schatz
'           Earth and Atmospheric Sciences Department
'           University of Alberta, 1998
'
' Date:  December 30th, 1998
'
' Title: Create New View
'
' Topics: View.Make, MultiInput, SetName, SetCreator, SetUnits, SetDistanceUnits, GetWin, Open
'
' Description: This Avenue Script is one in a series of scripts used to create the A R C L A extension. This script creates
a
'             new View Document within the current Project.  It also asks the user to input certain variables: View Name,
'             Creator Name and it automatically sets the Distance Units and Map Units to meters.
'
' Requires: A Project
'
' Self:
'
' Returns:  A new View Document
'
' Updated:  January 23rd, 1999
theProject = av.GetProject
'****
'Create a list of the elements first
'****
theViewElements = {"View Name", "Creator Name"}
theDefaultValues = {"View_1", "A R C L A user"}
'****
'Create a View Name and set other properties available from
'the list above
'****
theNewViewProperties = MsgBox.MultiInput("Please Input Values for the following information:", "Input View
Properties", theViewElements, theDefaultValues)
if (theNewViewProperties.Count = 0) then
    MsgBox.Error("No VIEW properties entered: script will end" + NL + "Program Cannot continue" + NL + "Press
OK to EXIT", "Warning")
    exit
end
'****
'Create the View
'****
theNewView = View.Make
if (nil = theNewView) then
    MsgBox.Error("No VIEW created: script will end" + NL + "Program Cannot continue" + NL + "Press OK to
EXIT", "Error")
    return(nil)
end
'****
'Obtain the user values
'****
theViewName = theNewViewProperties.Get(0)
theCreatorName = theNewViewProperties.Get(1)
'****
'Set the user values to the View document
'****
theNewView.SetName(theViewName.AsString)
theNewView.SetCreator(theCreatorName)
'****
'Open the new Document Window (View)
```

```
'****
theViewWindow = theNewView.GetWin
theViewWindow.Open
```

```
' Name:  View.RandomSelection
'
' Author: ESRI
'
' Modified by: Rod Schatz
'                    Department of Earth and Atmospheric Sciences
'                    University of Alberta, 1999
'
' Date:  January 29th, 1999
'
' Title:  Selects a percentage of a theme's features randomly
'
' Topics:  Analysis
'
' Description: This Avenue Script is one in a series of scripts used to create the A R C L A extension. This
'                     script will create a random selection of features based on a percentage entered by the user of
'                     the total number of features in the selected Theme.
'
' Requires:  A View document with at least one active theme.
'
' Self:
'
' Returns:
'
' Updated:
theView = av.GetActiveDoc
theDisplay = theView.GetDisplay
ThemeList = av.GetActiveDoc.GetActiveThemes
if (ThemeList.Count <> 1) Then
    msgbox.Error("You must have one and only one active theme", + NL + "Program Cannot continue" + NL + "Press
OK to EXIT", "Error")
    exit
end
theTheme = ThemeList.Get(0)
theFtab = theTheme.GetFtab
theBitMap = theFtab.GetSelection
theBitMap.ClearAll
TotalRecs = theFtab.GetNumRecords
PercentSelect = MsgBox.Input("Enter the percentage of Total Population:", "Select Random Sample","10")
if ((PercentSelect = NIL) or (PercentSelect.IsNumber.Not)) then exit end
if((PercentSelect.AsNumber > 99) or (PercentSelect.AsNumber < 1)) then
    Msgbox.Error("You must enter a number between 0 and 100", + NL + "Program Cannot continue" + NL + "Press
OK to EXIT", "Error")
    exit
end
selectnumber = (PercentSelect.AsNumber/100 * TotalRecs).Truncate
for each n in 1..selectnumber
    newnumber = Number.MakeRandom ( 0,  totalrecs)
    theBitMap.Set(newnumber)
end
theDisplay.Invalidate(True)
```

```
' Name: View.SetLambertProjection
'
' Author: Rod Schatz
'          Earth and Atmospheric Sciences Department
'          University of Alberta, 1998
'
' Date:  December 30th, 1998

'
' Title: Set View's Projection
'
' Topics:  View, Theme, FTheme, Project
'
' Description: This Avenue Script is one in a series of scripts used to create the A R C L A extension.  This
'              script changes the a View's Projection to Lambert Conformal Conic with a spheroid of GRS 80.
'              The script sets distance units and map units to meters.  This script is designed to work
'              directly with the 1996 Census Canada Shapefiles.
'
' Requires: A view with one or more Themes.
'
' Self:
'
' Returns:
'
' Updated: January 13th, 1999
theView = av.GetActiveDoc
'****
'Error Handling
'****
if (theView.Is(View).Not) then
    MsgBox.Error("Active Document is not a View", "Error")
    return nil
end
'****
'set the parameters of the Lambert Conformal Conic projection
'****
theNewProjection = Lambert.Make(theView.ReturnExtent)
theNewProjection.SetSpheroid(#SPHEROID_GRS80)
'****
'set Map Units to feet and Distance Units to miles
'****
theView.SetUnits(#UNITS_LINEAR_METERS)
theView.GetDisplay.SetDistanceUnits(#UNITS_LINEAR_METERS)
'****
'apply the projection to the view
'****
theView.SetProjection(theNewProjection)
```